
Unicenter

TCPaccess Telnet Server System Management Guide

Version 6.0



Computer Associates
The Software That Manages eBusiness



This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc. (CA)

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

Chapter 1: Introduction

Operations	1-1
System Commands	1-2
Command Scripts	1-2
Using the System Management Facility (SMF)	1-2
Diagnostic Procedures	1-3
TSO Diagnostic Commands	1-3

Chapter 2: Operation

Operator Commands	2-2
Starting and Stopping	2-2
Startup	2-2
Shutdown	2-3
Starting and Stopping Task Groups	2-3
Shutting Down with P versus P CLEAR	2-4
Recycling Task Groups	2-4
Related VTAM Commands	2-5
JCL Requirements	2-5
JCL Descriptions	2-5
Sample JCL for Starting Unicenter TCPaccess Telnet Server	2-6
Startup JCL Customization	2-7
Starting and Stopping Unicenter TCPaccess Telnet Server	2-8
T01LOG Log Spin Function	2-8
STARTxx Configuration	2-9
Initial STARTxx Customization	2-9
System Commands	2-9
Command Format	2-10
Subsystem Recognition Character	2-10
Task Group Identifier (TGI)	2-11
Verbs	2-11

Objects.....	2-12
Operands	2-12
Comments.....	2-13
Delimiters	2-14
General Task Group Commands	2-14
SNAP.....	2-14
STATUS.....	2-15
ARM Command.....	2-15
REFRESH Command	2-16
REFRESH	2-16
IJT Commands	2-17
GTF	2-17
HELP	2-18
IFS	2-18
LOGGING.....	2-18
MODULE.....	2-21
MEM.....	2-21
MVS	2-22
P	2-23
POOL.....	2-23
SET	2-25
SRC	2-26
START	2-26
STCK.....	2-27
STOP.....	2-27
SVCDUMP.....	2-28
TASK	2-29
TIME.....	2-29
TRACE	2-29
VSM.....	2-30
Command Scripts	2-30
Sample Command Script.....	2-31
Notes	2-32

Chapter 3: Using the System Management Facility (SMF)

Standard MVS SMF Record Header	3-2
Standard SMF Record Header	3-2
Descriptor	3-2
Product.....	3-3
Server Application Event Records	3-4

Task Identification Section	3-4
User Identification Section	3-4
Telnet Server Record (Subtype 23) Data Section	3-4
Access Method and Protocol Layer Event Records	3-6
Interval Records	3-6
Virtual Storage Statistics (Subtype 80) Data Section	3-6
SMF Report Writer Program	3-9
Commands	3-9

Chapter 4: Diagnostic Procedures

Problem Determination	4-1
Failures That Produce Dumps	4-2
Failures Due To Network Problems	4-2
Failure of Unicenter TCPaccess Telnet Server to Respond	4-2
Failure of Major Applications	4-2
Loops	4-2
Storage shortages	4-3
Problem Reporting	4-3
ABENDs	4-4
Incorrect Output	4-4
Startup and Parameter Errors	4-4

Chapter 5: Diagnostic Commands

STROBE	5-1
TCPEEP	5-2
User Interface	5-2
Viewing Trace Data	5-3
Trace Operation	5-3
TCPEEP Syntax	5-3
Useful TCPEEP Commands	5-7
TCPEEP Example	5-8
TRACE	5-10
MVS Component Trace	5-10
Trace Address Space	5-10
Exit	5-10
External Writer	5-11
Component Trace JCL	5-11
TRACE Command	5-12
TRACE Command Reply	5-13

TRACE Command Reply Options	5-15
Comparing TCPEEP and MVS TRACE Syntax	5-17
Trace Command Examples	5-17
Processing Trace Data	5-19

Index

Introduction

The System Management Guide provides information on maintaining Unicenter TCPaccess Telnet Server.

The organization of the guide is as follows:

- [Operations](#) – Describes operator commands, system, general task group, dynamic configuration, APP and IJT commands, JCL requirements, the STARTxx member of the PARM library, and command scripts.
- [Using the System Management Facility \(SMF\)](#) – Describes the SMF records, provides samples of the record headers and other records and an example of the SMF report writer program.
- [Diagnostic Procedures](#) – Describes the available diagnostic utilities to troubleshoot various Unicenter TCPaccess Telnet Server failures and the documentation needed to report problems to customer support.
- [Diagnostic Commands](#) – Describes how to troubleshoot, and describes the diagnostic commands.

Operations

Unicenter TCPaccess Telnet Server can be run as a batch job or as a started task. The chapter “Operation” describes Unicenter TCPaccess Telnet Server startup and shutdown.

The operator commands to start and shutdown Unicenter TCPaccess Telnet Server are described, including orderly shutdown and fast shutdown, shutting down with **P** versus **P CLEAR**, and recycling of task groups.

JCL is provided on the distribution tape, and this chapter provides a table of statements and description of the associated Unicenter TCPaccess Telnet Server JCL. Sample startup JCL and customization information is also given.

The T01LOG log spin function lets you specify that the T01LOG SYSOUT file be closed and re-opened on a regular basis determined by either the number of lines, a period of time, or by a combination of the two.

The START_{xx} configuration member is described and customization tips are provided.

System Commands

The format of system commands and some of the most commonly used commands are described.

The following task groups and their commands are described along with syntax examples:

- General Task Group Commands
- Dynamic Configuration Commands
- APP Task Group Commands
- IJT Task Group Commands

Command Scripts

Command scripts are a prearranged executable sequence of IFS task group commands and command script statements. A sample command script is provided and syntax is described.

Using the System Management Facility (SMF)

This chapter provides sample of the following SMF record formats:

- Standard MVS SMF record headers
- Standard Unicenter TCPaccess Telnet Server SMF record headers
- Server application event records
- Access method and protocol layer event records
- Interval records
- SMF report writer program

Diagnostic Procedures

This chapter provides information to help with problem determination and problem reporting.

Problem reporting tells you about ABENDs, incorrect output for Telnet, and hardware and startup and parameter errors.

TSO Diagnostic Commands

This chapter provides tips for troubleshooting and describes the diagnostic commands.

This chapter describes the basic operation of Unicenter TCPaccess Telnet Server.

It includes the following topics:

- [Operator Commands](#) – Describes the operator commands, including starting and stopping and related VTAM commands
- [JCL Requirements](#) – Describes the JCL issues for Unicenter TCPaccess Telnet Server operation and how to execute Unicenter TCPaccess Telnet Server as a started task
- [STARTxx Configuration](#) – Describes information for the STARTxx member
- [System Commands](#) – Describes functions, syntax, and arguments of Unicenter TCPaccess Telnet Server system commands
- [General Task Group Commands](#) – Describes the SNAP and STATUS commands
- [REFRESH Command](#) – Describes the REFRESH command that is processed by the APP task group
- [IJT Commands](#) – Describes the commands processed by the job step (IJT) task group
- [Command Scripts](#) – Describes how to build command scripts and includes a sample command script and some notes

Operator Commands

This section describes the operator commands available to Unicenter TCPaccess Telnet Server users.

Starting and Stopping

Unicenter TCPaccess Telnet Server can be run as a batch job or as a started task. Many installations route started JCL output to a purge output class.

Running Unicenter TCPaccess Telnet Server as a batch job (as opposed to a started task) can provide valuable information in the JES logs if problems arise. Due to cross-memory services restrictions in MVS, when Unicenter is run as a batch job, the JES initiator terminates when the job is stopped. For this reason it is probably better to run Unicenter TCPaccess Telnet Server as a started task. Use the appropriate start command to start the Unicenter TCPaccess Telnet Server JCL procedure.

Orderly shutdown of Unicenter TCPaccess Telnet Server notifies the application layer facilities that a shutdown was requested. These facilities can then terminate prior to the termination of the underlying protocols. Orderly shutdown also lets the task groups terminate gracefully with regard to the various interdependencies and interfaces that exist between them.

Startup

Unicenter TCPaccess Telnet Server starts by executing IFS. IFS initializes the address space as a whole and then starts the other task groups. To do so, IFS reads and executes the start command script defined by CMND=STARTxx in the PARM field of the EXEC statement and contained in the PARM data set defined by the SYSPROC DD statement.

Each task group is started by an IFS START command in the command file. This command, in turn, specifies the configuration file member containing the parameters used by the task group as it starts. The main configuration member for the task group is specified by CNFG(xx) on the IFS START command for the task group. Some task groups also use secondary configuration members from the PARM data set as specified in their main configuration file. Typically, a task group dynamically loads various other programs and tables from the STEPLIB data set during initialization and later, as needed.

Note: The IJT task group uses IJTCFGxx, since this task group controls all other task groups. Therefore, you cannot restart this task group without stopping and restarting the entire Unicenter TCPaccess Telnet Server address space.

Shutdown

You can initiate shutdown using these methods:

- Issue the MVS STOP command (**P**) for the Unicenter TCPaccess Telnet Server job name (for example, **P TCPaccess**)
- Use the IFS subsystem P command (subsystem recognition character preceding P).
- Use the MVS MODIFY command (for example, **F TCPaccess,P**)

The MVS operator is prompted after the STOP command to confirm the shutdown operation.

Note: See the NOPROMPT command in the *Customization Guide* to turn off the WTOR prompt.

- Depending on the number of worker tasks, open sockets, etc., the Telnet Server may take a minute to come down. (You can check the T01LOG DD to see the messages that are being produced during termination.) If time passes and the server appears to be hung:
 1. Issue **F TCPaccess, SVCDUMP**.
 2. When the SVCDUMP is complete, issue **C TCPaccess** to cancel the address space
 3. Contact technical support, supplying the SVCDUMP and address space output for problem resolution.

Starting and Stopping Task Groups

The individual APP task groups within Unicenter TCPaccess Telnet Server can be stopped and started independently with the IFS subsystem STOP and START commands.

Example

If the APP task group terminates abnormally, you can restart it without terminating and restarting the rest of Unicenter TCPaccess Telnet Server. See the section [START](#) for more information on restarting task groups.

Note: Any APP task group can be stopped and started to pick up new configuration information.

Shutting Down with P versus P CLEAR

The normal method of shutting down Unicenter TCPaccess Telnet Server is to issue the MVS STOP (P) command.

f TCPaccess,P

or

%P (substituting your subsystem identification character for %)

If you plan to install maintenance on the Unicenter TCPaccess Telnet Server before restarting Unicenter TCPaccess Telnet Server, use the following command to stop Unicenter TCPaccess Telnet Server:

f TCPaccess,P CLEAR

or

%P CLEAR (substituting your subsystem identification character for %)

In addition to normal STOP (P) processing, the P CLEAR command clears modules and control blocks from CSA and the subsystem hooks installed by this address space.

Note: Use the CLEAR option to stop Unicenter TCPaccess Telnet Server before applying the updates. This command also lets Unicenter TCPaccess Telnet Server use the most recent versions of the base modules when it is restarted.

When a P CLEAR is issued, the Telnet Server may take an SOC1 or SOC4 during termination. This is normal due to the missing CSA control blocks and modules. Only one P CLEAR is needed to clear the CSA.

After issuing a P CLEAR, it may be necessary to follow additional stopping instructions for Unicenter TCPaccess Telnet Server. For more information, see the section [STOP](#)

Recycling Task Groups

While such action is typically not needed, individual APP task groups can be stopped and restarted while Unicenter TCPaccess Telnet Server is running.

To stop an APP task group, issue the following command from the MVS operator console:

F TCPaccess,STOP APP TASK(nn)

To restart a task group, issue the following command from the MVS operator console

```
F TCPaccess,START APP CNFG(nn)
```

One or more IFS START commands can be issued from a command script as shown in the startup example in [Startup JCL Customization](#).

Related VTAM Commands

To activate the major node to VTAM for the Telnet Server, issue the following command:

```
V NET,ACT,ID=major_node_name
```

(The distributed major node is A03ACCES.) This command also activates the LUs required for Server Telnet.

JCL Requirements

The JCL provided on the distribution tape is functionally equivalent to that listed in the following topics, but may differ in the order of JCL statements or the contents of comment statements included in the JCL. (See the following table, in [JCL Descriptions](#), which contains statements and descriptions of the Unicenter TCPaccess Telnet Server JCL).

Note: The sample startup JCL, RUNTLN, is in *TRGINDX.CNTL*(RUNTLN).

JCL Descriptions

The following table lists Unicenter TCPaccess Telnet Server JCL DD statements and descriptions.

DD Statement	Description
ABNLIGNR	Disables the Abend-Aid program product.
ARPAHELP	Defines a PDS whose members contain HELP text.
STEPLIB	Defines the load library for Unicenter TCPaccess Telnet Server. These data sets must be APF-authorized.
SYSHELP	Defines the help data set for the Unicenter TCPaccess Telnet Server task groups.

DD Statement	Description
SYS Parm	Defines the PARM data set that contains all the configuration file members, which provide the parameters for the various task groups within Unicenter TCPaccess Telnet Server.
SYS PROC	Defines the PARM data set from which command scripts (command file members) are obtained by the IJT task group. In particular, the PARM data set contains the command file member STARTxx, which contains the commands to start Unicenter TCPaccess Telnet Server.
SYS DUMP	Contains formatted system dumps.
T01LOG	Defines the log file.

Sample JCL for Starting Unicenter TCPaccess Telnet Server

Use this Unicenter TCPaccess Telnet Server JCL with all interfaces including LOOPBACK. Edit the appropriate member, supplying the correct start member and verifying other symbolic parameters for accuracy.

```
//RUNTLN JOB
//*
//*   Sample JCL procedure to run the TLN Server.
//*
//*   Edit the TRGINDX, SSN, SRC, SOUT, CMND symbolic
//*   parameters.
//*
//*   Verify that the JOB card and naming conventions meet
//*   your site's JCL requirements, then submit this JOB.
//*
//TLN PROC TRGINDX='TRGINDX', TARGET LIBRARIES DSN INDEX
//          SSN=ACTN,          DFLT SUBSYSTEM NAME
//          SRC='% ',          DFLT SUBSYSTEM RECOGNITION CHAR
//          SOUT='* ',         CHOOSE A HOLD NONPURGE SYSOUT CLASS
//          CMND=STARTTN       DFLT STARTUP COMMAND SCRIPT NAME
//          CNFG=TN            IJTCFGxx SUFFIX
//*
//TCP/IP EXEC PGM=IFSSTART,REGION=6144K,TIME=1440,
// PARM='IFSINIT,U=&SSN,SR=&SRC,SO=&SOUT,CM=&CMND,CF=&CNFG'
//*
//STEPLIB DD DISP=SHR,DSN=&TRGINDX..LOAD
//          DD DISP=SHR,DSN=GSK.SGSKLOAD
//*
//* WARNING: THE LOAD DATA SET MUST NEVER BE ADDED TO THE LINK LIST.
//*          TCPACCESS' ELEMENT NAMES ARE NOT UNIQUE AND COULD AFFECT
//*          THE OPERATIONS OF OTHER SOFTWARE. THE LOAD DATA SET
//*          SHOULD ALWAYS BE REFERENCED THROUGH A STEPLIB OR JOBLIB
//*          STATEMENT.//*
//*          CONFIGURATION DATA SETS
//*
//SYS Parm DD DISP=SHR,DSN=&TRGINDX..PARM
//SYS PROC DD DISP=SHR,DSN=&TRGINDX..PARM
//*
```

```

/**      LOG DATA SETS
/**
//T01LOG DD  SYSOUT=&SOUT
/**
/**      DUMP DATA SETS
/**
//SYSUDUMP DD  SYSOUT=&SOUT
/**
/**      MISC DATA SETS
/**
//ARPAHELP DD  DISP=SHR,DSN=&TRGINDX..HELP
//SYSHelp DD  DISP=SHR,DSN=&TRGINDX..HELP
//ABNLIGNR DD  DUMMY                      /* DISABLE ABEND-AID PROCESSING */
//
//      PEND
//TLN      EXEC TLN

```

Startup JCL Customization

Make the following changes to the startup JCL on the PROC statement:

TRGINDX= Enter the high level qualifier of the Unicenter TCPaccess Telnet Server data sets.

SSN= Enter the subsystem ID you are using.
Alias: U.
Default: ACSS.

SRC= Enter the subsystem recognition character you are using. (Verify that it does not conflict with any other subsystems you are running.)
Alias: SUBS.
Default: %.

SOUT= Choose an installation defined SYSOUT class that will be held and not automatically purged at job or started task termination.
IFS uses this SYSOUT class for dynamically allocated SYSOUT data sets produced during task ABENDs, output from the operator SNAP command, and task initialization to print copies of the task startup PARM members.
Alias: SYSO.
Default: *.

CMND= Enter the name of the STARTxx member in the PARM library.
Alias: CMD.
Default: START00.

CNFG= Enter the suffix of the IJTCTGxx configuration member.
Alias: CFG.
Default: 00.

CAUTION! Many installations route started task output to a purge class for deletion at started task termination. If an installation's default SYSOUT class is purged at job or started task termination and is using SOUT='*', then all dumps or SNAPs produced will be lost.

Note: If you have not link-listed the Unicenter TCPaccess Telnet Server LINK library, you must not reference it in the startup JCL. However, if you use any of the client commands such as TCPEEP, you need to STEPLIB to it in the TSO logon procedure (or JCL, if you run the commands in batch).

These are the only changes required. Copy the JCL stream from TRGINDX.CNTL(RUNTLN) to your started task procedure library.

You are now ready to start Unicenter TCPaccess Telnet Server. Before doing so, it is a good idea to review the installation steps in the Unicenter TCPaccess Telnet Server *Getting Started Guide* to ensure that you have made all the necessary changes.

Starting and Stopping Unicenter TCPaccess Telnet Server

To start Unicenter TCPaccess Telnet Server, issue the MVS **START** command.

To shut down the address space you can use either an MVS MODIFY command (F *TCPaccess,P*), the subsystem recognition character for example, %P), or an MVS STOP command (**P TCPaccess**).

The following system message appears requesting confirmation:

```
T01IF013R Confirm request to stop A/S -- Reply 'Y' or 'N'
```

As termination continues, messages are issued indicating that various components are terminating.

T01LOG Log Spin Function

The T01LOG log spin function lets you specify that the T01LOG SYSOUT file should be closed and reopened on a regular basis determined by the number of lines, a period of time, or by a combination of the two. The log spin function is implemented by including the necessary spin parameter on the LOGGING statement in the IJTCFGxx member. See the *Customization Guide* for information about the IJTCFG member.

This feature is useful for customers who have long uptimes where spool usage can grow to the point that it impacts Unicenter TCPaccess Telnet Server performance.

This feature, along with the addition of FREE=CLOSE on the T01LOG DD statements, lets you examine, print, or purge output at the convenience of the installation and prevent uncontrolled growth of the spool.

If you specify FREE=CLOSE on the T01LOG DD statement, it causes the SYSOUT data set to be freed and made available on the output queue. Without this DD parameter, at the closing of the T01LOG file, the output remains as part of the total job output with each iteration of output appended to the last.

Note: There is currently no ability to spin the logs at a specified time of day (for example, at 8 A.M. each day). You can only specify hours since Unicenter TCPaccess Telnet Server was started.

STARTxx Configuration

The STARTxx member in the PARM library is an IFS command script that tells Unicenter TCPaccess Telnet Server which task groups to start and which configuration members to use.

The STARTxx member is the highest-level configuration file, since it points to the configuration members for each of the task groups. It is pointed to by the CMND= symbolic parameter in the startup JCL procedure.

The default file, START00, as distributed, contains the following:

```
DISPLAY IFS
DISPLAY SRC
START APP CNFG(TN)
SET TEST ON TGB(IJT)
```

Initial STARTxx Customization

Individual commands are described later in this chapter. Do **not** eliminate any of them.

1. Make a copy of the START00 member, giving it a new name (such as START01).
2. Modify the JCL startup procedure and point to your updated STARTxx member.

System Commands

This section describes, in reference form, the functions, syntax, and arguments of all the Unicenter TCPaccess Telnet Server system commands.

The most commonly used commands are:

TASK	Displays active task groups.
STATUS	Displays maintenance status of a task group.
P CLEAR	Removes system hooks at shutdown.
STOP APP TASK(nn)	Stops the task group (indicated by <i>TGI</i>).
START APP CNFG (<i>xx</i>)	Starts the task group <i>TGI</i> using CNFG(<i>xx</i>), where <i>xx</i> is the task group identifier for the task group you are starting.
REFRESH LUPARM (APPLUP <i>xx</i>)	Refreshes LUPOOL and LU information for Server Telnet usage.

Command Format

A command consists of a Subsystem Recognition Character (SRC), optionally followed by a Task Group Identifier (TGI), followed by a command verb, a command object, and usually, by one or more operands. Commands are referred to by the command object. These terms are described in the sections that follow. This examples uses % as the SRC:

%P CLEAR

You can also use these commands with the MVS modify command, specifying the job name. For example:

F TCPaccess,P CLEAR

Note: Commands are limited to 126 characters.

Subsystem Recognition Character

The SRC provides the method for an MVS subsystem address space to have operator commands directed to it. The person responsible for installing the subsystem sets the SRC, and it is specified in the RUNTLN job to start Unicenter TCPaccess Telnet Server. It is one of the parameters used in the execute step for IFSSTART.

The SRC can be any valid SRC supported by JES2 or JES3 (JES2 uses \$, JES3 uses *). Ask your JES systems programmer which character to use and ensure that it is not a character used by an installation subsystem. The *JES2/JES3 Initialization and Tuning* documentation defines valid SRCs to use when passing commands from local consoles to subsystems (see CONDEF statement, CONCHAR argument, for JES2; and CONSTD statement, SYN argument, for JES3).

SRC is an optional parameter. If no SRC is specified, there will not be an SRC (and all commands issued to the Unicenter TCPaccess Telnet Server address space must be done via the MVS MODIFY command).

Task Group Identifier (TGI)

Most commands are processed by an implied task group depending on the command object. Some commands provided by Unicenter TCPaccess Telnet Server, such as SNAP, can be directed to a specific task group; in this case, the task group identifier must be placed between the SRC and the verb.

Use the following command to direct the SNAP command to the APP task group:

```
APP SNAP ALL
```

If the task group identifier is omitted, the SNAP command, entered as **SNAP**, is directed to the job step task group (IJT). The keyword ALL indicates to SNAP the IFS trace table.

The following is a list of valid task group identifiers:

APP	TCP/IP applications (i.e., the Telnet Server).
IJT	IFS Jobstep Task.

Verbs

The following is a list of command verbs and the action taken for each verb:

Note: If you do not specify one of the following verbs, DISPLAY is assumed.

DISPLAY	Displays the status of the command object.
MODIFY	Modifies or changes the value of the command object. The keyword ADD, DELETE, or UPDATE usually appears as an operand in this command.
VARY	Changes the status of the command object in an orderly manner. The keyword ON or OFF usually appears as an operand in this command.

Some command objects, such as SNAP, support only the DISPLAY form and ignore any other specified verb.

You can enter verbs spelled exactly as they are shown or you can use an acceptable abbreviation. You can abbreviate any verb by entering only the significant characters. That is, you must type as much of the verb as is necessary to distinguish it from other verbs. DISPLAY, MODIFY, and VARY can be abbreviated as D, M, and V, respectively.

Objects

Verb action is performed on command objects. SNAP is a command object. The SNAP command can be entered in these ways:

```
DISPLAY SNAP
SNAP
```

You can enter command objects spelled exactly as they are shown or you can use an acceptable abbreviation. You can abbreviate any object by entering only the significant characters. That is, you must type as much of the object as is necessary to distinguish it from other objects.

Operands

Operands provide the specific information required for the command to perform the requested operation.

These operands for the SNAP command specify either a JES SYSOUT class or, alternately, an MVS SNAP:

```
SNAP CLASS(A)          /* JES SYSOUT CLASS      */
SNAP MVS                /* MVS SNAP          */
```

Note: These examples issue the SNAP against the IFS Jobstep Task Group (IJT) by default.

The types of operands used with commands are:

- Positional
- Keyword

Positional Operands

Positional operands follow the command object in a prescribed sequence.

You must replace the TGI with the actual three-character task group identifier when you enter the command.

When you want to enter a positional operand that is a list of several names or values, the list **must** be enclosed within parentheses. The names or values must not include unmatched parentheses.

Keyword Operands Keywords are specific names or symbols that have a particular meaning to the system. You can include keywords in any order following the positional operands. In the command descriptions within this document, keywords are shown in uppercase characters. A typical keyword is ADD.

Some keywords let you specify values. Place the value inside parentheses following the keyword.

The following is a typical keyword with a value:

```
TG ( TGI )
```

You select the task group identifier desired and substitute that value for *TGI* when you enter the operand, for example:

```
TG ( IJT )
```

Note: If conflicting keywords are entered, the last keyword entered overrides the previous ones.

**Abbreviating
Keyword Operands** You can enter keywords spelled exactly as they are shown or you can use an acceptable abbreviation.

You can abbreviate any keyword by entering only the significant characters. That is, you must type as much of the keyword as is necessary to distinguish it from the other keywords of the command object.

The SNAP command includes the keywords CLASS and MVS.

- Abbreviations for CLASS are C, CL, CLA, and CLAS
- Abbreviations for MVS are M and MV

In addition, some commands allow unique abbreviations or aliases for some of their keywords.

Comments

Comments can be added to a command anywhere a blank might appear. Enter them within the comment delimiters, */** and **/* as shown below:

```
SNAP C      /* OVERRIDE DEFAULT SYSOUT CLASS */
```

Delimiters

When you type a command, you must separate the command task group identifier, verb, object, and operands from each other using one or more blanks or a comma.

```
POOL ( SRB XWA )
```

Note: Do not use a semicolon as a delimiter. Characters entered after a semicolon are ignored.

General Task Group Commands

Unicenter TCPaccess Telnet Server general task group commands can be processed by any active task group in a Unicenter TCPaccess Telnet Server address space. If the task group identifier is not specified, the command is processed by the job step task group (IJT). The commands are SNAP and STATUS.

SNAP

Use the SNAP command for debugging purposes to spin off (dynamically allocate and free) a SYSOUT data set containing a formatted snap dump of control blocks for a task group:

```
[ TGI ] SNAP [ MVS ] [ ALL ] [ CLASS ( SYSOUT_class ) ]
```

TGI Specifies the three-character task group identifier of the task group to process the command. If not specified, IJT is assumed.

You can also append the task number to the task group identifier. For example, to snap the control structures for APP task group, task #2:

```
(CellCmdEnv)APP2 SNAP ALL CLASS(X)
```

MVS Specifies an MVS formatted snap dump instead of a Unicenter TCPaccess Telnet Server formatted snap dump.

.

ALL Include the Unicenter TCPaccess Telnet Server internal trace table in a Unicenter TCPaccess Telnet Server formatted snap dump. Applications can also include extra information.

CLASS(SYSOUT_class) Specifies an override value for the SYSOUT class to dynamically allocate.

The default class is specified by the SOUT= symbolic parameter in the JCL procedure for this address space.

Examples

The following are examples of the SNAP command:

```
SNAP
SNAP MVS
SNAP ALL CLASS( A )
APP2 SNAP ALL CLASS(X)
```

STATUS

Use the STATUS command to display the maintenance status of a task group. The version and release numbers are displayed.

```
[ TGI ] STATUS
```

Examples

The following are examples of this command:

```
STATUS
APP STATUS
```

ARM Command

ARM *request*

Address space operator command.

The valid *request* types are:

START	Synonym for the ENABLE, LISTEN, REGISTER, and WAITPRED commands, issued in that order.
ENABLE	Enables ARM processing for the address space, and allows the issuing of other commands.
LISTEN	Requests the address space to listen for ARM-related ENF events.
REGISTER	Registers the address space as an ARM element in the sysplex.
WAITPRED	Requests that the address space wait for its ARM predecessors to become ready.
READY	Makes the address space eligible for restart by ARM.

STOP	Synonym for the DISABLE, NOLISTEN, and DEREGISTER commands, issued in that order.
DISABLE	Disables ARM processing for the address space. When it completes, any further ARM-related ENF events are ignored.
NOLISTEN	Requests the address space to stop listening for ARM-related ENF events.
DEREGISTER	Deregisters the address space as an ARM element in the sysplex.

Normally, you will not need to issue ARM commands – IFS will perform the required ARM actions automatically.

REFRESH Command

This section describes the REFRESH command

REFRESH

Use the REFRESH command to refresh certain configuration parameters of the APP task group. It can be used to refresh the LU pool, greeting member, or the USS Table.

REFRESH TASK(*n*) [LUPARM(*mem_name*) | GREETING(*mem_name*) | USSTAB(*tbl_name*)

TASK (*n*) Specifies the task number of the task within the APP task group to which the command is directed.

Default: Task number one.

LUPARM(*mem_name*) The LU pool is refreshed from this member.

GREETING(*mem_name*)

Specifies the name of the member of the ARPAHELP data set from which the new server telnet greeting will be read.

Subsequent Telnet sessions will be presented with the greeting found in this member.

USSTAB (*tbl_name*) Specifies the USS table (*tbl_name*) coded in the active APPCFGxx SERVICE segment should be refreshed.

Note: LUPARM, GREETING, and USSTAB are mutually exclusive.

Examples

```
REFRESH LUPARM ( APPLUP00 ) TASK ( 2 )
APP REFRESH TASK ( 1 ) GREETING( GREETING )
REFRESH USSTAB ( T01USS01 ) TASK ( 1 )
```

IJT Commands

This section describes commands processed by the job step (IJT) task group.

GTF

Use the GTF command to display or modify the settings of GTF trace event flags. A trace event is recorded only when an event is turned on (with this command) and the task group executing a module that invokes a trace event is in GTF mode (for more information, see [SET](#)). If there is no job name the trace applies only to the Telnet Server address space.

```
[ DISPLAY | MODIFY ] GTF [ ON | OFF ]
[ EI ( event_id [ ... ] ) | CB ( cb_id [ ... ] ) | MOD ( mod_name [ ... ] ) | ALL
]
```

DISPLAY | MODIFY Specifies whether to display or modify the settings of GTF trace event flags.

ON | OFF Select only those events that are turned on or off for display, or to turn on or off a specified event(s) for modify.

If neither ON nor OFF is specified with DISPLAY, the on/off state of an event is not considered for inclusion in the display.

Note: Either ON or OFF is required with MODIFY.

EI (*event_id*) Select the event identifiers listed (one- to eight-alphanumeric characters).

CB (*cb_id*) Select the events for the control block identifiers listed (one- to four-alphanumeric characters).

MOD (*mod_name*) Select the events generated by the modules listed (one- to four-alphanumeric characters).

Note: A name stem is permitted. That is, if IFSP is specified, all modules having names starting with IFSP are included.

ALL Select all events.

Examples The following are examples of this command:

```
GTF OFF
GTF EI( MESSAGE CALLPC SRBDISP )
GTF CB( SSOB ISRB )
MODIFY GTF OFF ALL
MODIFY GTF ON CB( MODI SDWA )
DISPLAY GTF
```

HELP

Use the HELP command to get online information about the function, syntax, and operands of commands. This reference information is contained in the SYSHELP DD data set(s) and is displayed on your console in response to your request for help.

Enter HELP:

- Without operands to get an introduction to using the help facility
- With the operand COMMANDS to view a list of all the Unicenter TCPaccess Telnet Server commands for which help is available

```
HELP [ cmd_name | COMMANDS | GENERAL POOL ]
```

cmd_name Specifies the full name of a command for which help information is requested.

COMMANDS Requests a list of all commands for which help information is available.

GENERAL Requests a display of the general format and syntax of Unicenter TCPaccess Telnet Server commands.

POOL Requests a display of Unicenter TCPaccess Telnet Server pools.

Example The following are examples of this command:

```
HELP
HELP COMMANDS
HELP POOL
```

IFS

Use the IFS command to display environmental settings for the address space and to display key subsystem-related control block addresses.

IFS

This command has no arguments or keywords.

LOGGING

Use the LOGGING command to reparse and update the entire LOGGING statement, as contained in the IJTCFGxx startup member. Any parameter can be changed, within valid limits, and the entire statement can be reprocessed on an active gateway.

```
LOGGING [ CLASS ( class ) ]  
        [ DEST ( destination ) ]  
        [ NOW ]  
        [ PRINT ( subparm [ , subparm [ , ... ] ] ) ]  
        [ ROUTCDE ( list ) ]  
        [ SPIN ( LINES ( lines ) | MINUTES ( minutes ) | SYNC ) | NOSPIN ]  
        [ WTO ( subparm [ , subparm [ , ... ] ] ) ]
```

CLASS (*class*) Specifies the SYSOUT class.

Default: Class specified as SOUT= keyword of PARM field.

DEST (*destination*) Specifies the SYSOUT destination.

Default: No destination.

NOW When this parameter is issued through the console, an immediate logspin is performed.

Note: NOW works in conjunction with the SPIN parameter.

PRINT (*subparm*) Subparameters are processed left to right.

Valid values:

ALL WTO/PRINT all messages, all types.

NONE WTO/PRINT no messages.

(ALL,types) WTO/PRINT given types for all components.

(component,ALL) WTO/PRINT all messages for given component.

(component,NONE) WTO/PRINT no messages for given component.

(component,types) WTO/PRINT given messages for given component.

ROUTCDE (<i>list</i>)	Specifies the MVS routing codes for console messages.
<i>list</i>	One or more valid MVS routing codes, separated by commas. You specify routing code ranges by separating them with a hyphen.
	IFSPARM LOGGING ROUTCDE(2)
	IFSPARM LOGGING ROUTCDE(3,4,8-11)
	IFSPARM LOGGING ROUTCDE(9-11)
	Range: 1-16.
	Default: No routing code.
	Note: This means console messages are routed according to the defaults specified in the MVS SYSGEN.
SPIN (<i>LINES</i> (<i>lines</i>) <i>MINUTES</i> (<i>minutes</i> SYNC) NOSPIN	
	Determines when the log file closes and reopens.
<i>LINES</i>	Number of lines to be written to each log file before it is closed and reopened.
<i>MINUTES</i>	Duration of time before logout is done.
	Alias for <i>MINUTES</i> is <i>TIME</i> .
<i>SYNC</i>	Synchronizes to the hour.
	Default: <i>NOSPIN</i> .
WTO (<i>subparameter</i>)	Subparameters are processed left to right.
	Valid values:
<i>ALL</i>	WTO/PRINT all messages, all types.
<i>NONE</i>	WTO/PRINT no messages.
(<i>ALL</i> , <i>types</i>)	WTO/PRINT given types for all components.
(<i>component</i> , <i>ALL</i>)	WTO/PRINT all messages for given component.
(<i>component</i> , <i>NONE</i>)	WTO/PRINT no messages for given component.
(<i>component</i> , <i>types</i>)	WTO/PRINT given messages for given component.

You can also change logging dynamically with the MODIFY command. For example:

```
MODIFY job_name LOGGING parameter(s)
```

MODULE

Use the MODULE command to display information about a resident module such as call count and assembly date and time.

```
MODULE [ ( mod_name [ ... ] ) | ALL | * ]
```

mod_name [...] Module name(s) to display (one- to eight- alphanumeric characters).

ALL or * Displays all resident modules.

Examples The following are examples of this command:

```
MODULE *  
MODULE IFSSCALL IFSXPOST
```

MEM

Use the MEM command to display up to 1024 bytes of virtual storage. There are two versions of the command:

```
MODIFY MEM
```

```
DISPLAY MEM.
```

```
MODIFY MEM addr Vdata Rdata
```

addr Storage address to alter, expressed as a hex string.

Vdata Value of existing data at that location, expressed as a hex string.

Rdata Value of replacement data, expressed as a hex string

- Usage Notes
1. Storage in key zero cannot be modified.
 2. *add*, *Vdata*, and *Rdata* must contain an even number of characters 0-9, A-F. All three are mandatory.
 3. *Rdata* and *Vdata* can be 2 - 512 characters long, or 1 - 256 data bytes.
 4. *Rdata* may be longer than *Vdata*.

```
DISPLAY MEM addr | * [ DECLEN ( nnn ) | HEXLEN ( xxx ) MOD ( mod_name ) ]
```

addr or * Starting display address.

It can be entered as an explicit address or as an asterisk (*) with a module name (MOD(*mod_name*)) parameter

MOD(*mod_name*) Module name.

DECLEN | HEXLEN Length. It can be specified as decimal (DECLEN(*nnn*)) or hexadecimal (HEXLEN(*nnn*)), with a maximum of 1024.

If not specified, default value = 16.

Alias for DECLEN is LEN (abbreviated as DECL).

Note: HEXLEN has no alias (abbreviated as HEXL).

MVS

Use the MVS command to display the MVS environment and, optionally, the contents of selected control blocks.

MVS [IFS | JESCT | LNKST | SCVT | SMCA | SSCT]

IFS Displays the subsystem communication vector table address and name of each defined IFS-based subsystem.

JESCT Displays the address of the JES control table and the name of the primary JES.

LNKST Displays the names of the data sets in the MVS Link Library List.

SCVT Displays, in dump format, the MVS secondary communications vector table.

SMCA Displays, in dump format, the SMF Control Area.

SSCT Displays the subsystem communication vector table address and name of each defined subsystem.

Examples The following are examples of this command:

```
MVS
MVS SSCT
```

P

The P command terminates all task groups and the address spaces. Optionally, it removes the subsystem hooks installed by this address space at initialization.

P [CLEAR]

CLEAR

Specifies to clear the subsystem hooks installed by this address space before returning to MVS.

To update Unicenter TCPaccess Telnet Server after applying maintenance, use the CLEAR option to stop Unicenter TCPaccess Telnet Server before applying updates. In addition to providing normal stop processing, this also clears control blocks and certain modules in the CSA and lets Unicenter TCPaccess Telnet Server use the most up-to-date versions of the base modules when it is restarted.

Note: S0C1/S0C4 messages during termination after a P CLEAR are normal and can be ignored. Only one P CLEAR is needed to clear the CSA.

Examples

The following are examples of this command:

```
P
P CLEAR
```

POOL

Use the POOL command to display the statistics or attributes of data area pools. A pool is a collection of fixed-length data areas residing in a single MVS storage subpool managed by Unicenter TCPaccess Telnet Server without the overhead of GETMAIN/FREEMAIN.

POOL [(*pool_name* [...]) | *] [ATTR]

pool_name

Specifies the name of pool(s) to be displayed.

Options are:

FRR	IFS Recovery Element.
MB1	Buffer pool for moving data - 128 bytes.
MB2	Buffer pool for moving data - 384 bytes.
MB3	Buffer pool for moving data - 640 bytes.
MB4	Buffer pool for moving data - 1152 bytes.
MB5	Buffer pool for moving data - 1536 bytes.
MB6	Buffer pool for moving data - 2048 bytes.

MB7	Buffer pool for moving data - 5120 bytes.
MB8	Buffer pool for moving data - 10240 bytes.
MB9	Buffer pool for moving data - 16384 bytes.
MBA	Buffer pool for moving data - 32768 bytes.
MSRB	Message Service Request Block.
MWA	Module Work Area.
QCB	Queue Control Element for pools.
RTPB	TelnetRTM port block.
RTOB	TelnetRTM owner block.
SRB	IFS Service Request Block.
STAK	Module Stack Block for work areas.
XWA	Cross Memory Work Area.

ATTR

Specifies to display the pool attributes instead of statistics. Pool attributes are the values used to control expansion and minimum or maximum values. (Set by the POOLDEF configuration parameter statement or the default.)

Default values for the POOL options are described in the following table.

Value	Initial	Minimum	Expand	Contract
FRR	100	200	50	0
MB1	32	128	16	128
MB2	32	256	16	256
MB3	32	192	16	128
MB4	32	160	16	128
MB5	16	128	8	64
MB6	16	96	4	32
MB7	8	48	8	16
MB8	4	32	4	16
MB9	4	16	4	16

Value	Initial	Minimum	Expand	Contract
MBA	2	8	2	8
MSRB	200	400	100	0
MWA	100	200	50	75
QCB	100	200	50	0
RTPB	4	8	4	4
RTOB	32	256	64	128
SRB	100	200	50	0
STAK	40	20	20	0
XWA	112	160	24	0

Note: It is best to use the defaults at first, and issue the POOL command every so often to display pool usage. If you find pools being expanded and staying at the higher value, you can override the default and specify a higher minimum value.

Examples

The following are examples of this command:

```
POOL
POOL SRB ATTR
POOL * ATTR
```

SET

Use the SET command to set execution options for a task group.

```
SET [ DEMO | TEST | GTF ]
    [ ON | OFF ]
    [ TG( TGI [ ... ] ) | ALL ]
```

DEMO | TEST | GTF Specifies which mode to set:

DEMO	Some messages or processing is performed only in DEMO mode.
TEST	Some messages or processing is performed only in TEST mode. Some API and APP trace events are recorded only in TEST mode.
GTF	GTF events are recorded only when GTF mode is on.

ON | OFF Specifies to turn the specified mode on or off.

TG(*TGI* [...]) | ALL Specifies the three-character task group identifier affected by the mode change or ALL.

If not specified, all active task groups are affected.

Examples The following are examples of this command:

```
SET DEMO ON
SET GTF OFF TG( APP)
SET TEST OFF TG( IJT )
SET GTF ON
```

SRC

Use the SRC command to display or modify the subsystem recognition character for the address space.

```
[ DISPLAY ] SRC
MODIFY SRC [ char ]
```

char Specifies the new subsystem recognition character (one alphanumeric) for the address space.

The change is effective immediately. This must be a character acceptable by MVS console services. If no character is entered, there is no subsystem recognition character.

Examples The following are examples of this command:

```
SRC
MODIFY SRC #
DISPLAY SRC
```

START

Use the START command to start an APP task group and to specify initialization parameter overrides.

```
START [ APP CNFG( xx ) MEMBER( mem_name ) ]
```

TGI Specifies the three-character task group identifier of the task group to start.

CNFG(*xx*) Specifies the two-character suffix to use in constructing the name of the configuration parameter file.

The name takes the form TGICFGxx, where TGI is the task group identifier specified with the STARTxx command.

Note: If CNFG() is not specified, a value of 00 is assumed for xx.

The configuration parameter member resides in the SYSPARM DD data sets and provides site-specific configuration parameter values for the task group being started.

MEMBER(*mem_name*) Specifies the name of a command script to invoke when task group initialization completes (one- to eight-alphanumeric characters).

The command script resides in the SYSPROC DD data sets.

Examples The following is an example of this command:

```
START APP MEMBER( APPCFG05 )
```

STCK

Use the STCK command to convert the binary eight-byte clock value to a useful date and time.

STCK X'*hex_string*'

hex_string Specifies the eight-byte binary clock value as stored by the STCK instruction.

This value is expressed as a 16-character hex string within quotes.

Example The following is an example of this command:

```
STCK X'AF82198271FB4401'
```

STOP

Use the STOP command to terminate an APP task group. The task group performs an orderly shutdown before terminating. Static control blocks are left in a state for reuse if the task group is started again. Depending on the specific task group, the stop request can be delayed to let work in progress complete.

STOP APP TASK(*n*)

You can have up to four APP task groups active concurrently.

Example

```
STOP APP TASK( 1 )
```

SVCDUMP

Use the SVCDUMP command to generate a system formatted dump.

```
SVCDUMP [ ASID ( asid_list ) ]  
        [ JOBname ( jobname_list ) ]  
        [ GRPLIST ( group_list ) ]
```

ASID (*asid_list*)

A list of address space IDs. Unicenter TCPaccess Telnet Server includes these address spaces in the dump.

You can combine this parameter with the JOBname parameter for five address spaces.

The address space IDs can be listed in decimal or in hexadecimal notation, as x'999'.

JOBNAME (*jobname_list*)

A list of one- to eight-character job names.

Unicenter TCPaccess Telnet Server includes these address spaces in the dump.

You can combine this parameter with the ASID parameter for five address spaces.

Note: If there is more than one job active in the system with the same job name, Unicenter TCPaccess Telnet Server dumps the first one found in the Address Space Vector Table.

GRPLIST(*group_list*)

The name of one or more XCF groups whose member systems are included in the SVC dump.

This parameter allows SVC dumps to be requested for remote systems in the sysplex. When you specify this parameter, an SVC dump is captured on each system in the sysplex for which a member is active in one or more of the XCF groups specified.

The value specified must be from one- to eight-bytes in length and can be expressed as a quoted character string ('ABC') or a non-quoted character string where a separator or delimiter indicates the end.

The names specified can include wild card characters with the character ? (question mark) denoting a single mask character and the character * (asterisk) denoting zero or more mask characters.

TASK

Use the TASK command to display the active task groups. Information displayed includes execution-related flags, dispatch count, and the date/time of task group initialization.

TASK [(*TGI* [...])]

TGI

Specifies the three-character task group identifier of the task group to display.

Default: If *TGI* is omitted, all task groups are displayed.

Examples

The following are examples of this command:

TASK

TASK (APP)

TIME

Use the TIME command to display the current date and time in all useful forms. This command uses no parameters.

TRACE

Use the TRACE command to:

- Display current trace table status
- Turn internal tracing on or off
- Set the internal trace table size

The trace table is formatted and included in an IFS-formatted snap dump if an ABEND occurs.

**[DISPLAY] TRACE
MODIFY TRACE [ON | OFF] [SIZE(*number* | 16)] [FIXED]**

ON | OFF

Specifies to enable or disable internal tracing.

SIZE(*number*)

Specifies the size (*number*) of the trace table as a number of four KB (4096) pages (one or two decimal digits). Each entry in the trace table is 64 bytes long. The maximum number of four KB pages is 2048.

Use the SIZE parameter only when ON is also specified.

Default: 16 4 KB pages(64 KB).

FIXED

Specifies that the trace table be in fixed storage, allows trace capture of I/O-related events.

Normally this facility is not required, but you can request it for diagnostic purposes.

Examples

The following are examples of this command:

```
MODIFY TRACE OFF
TRACE
MODIFY TRACE ON
MODIFY TRACE ON SIZE( 8 )
```

VSM

Use the VSM command to display virtual storage usage statistics.

Note: This command uses no parameters.

Command Scripts

This section describes how to build command scripts that are a prearranged executable sequence of IFS task group commands and command script statements that you can invoke by specifying the command script name prefixed with a percent sign (%). The START xx PARM member is an example of a command script.

Command scripts are read from the SYSPROC DD data sets and a sample START00 script is provided in the PARM data set. You can invoke a command script at address space initialization by specifying the command script name with the CMND= parameter in the JCL procedure for an IFS address space. Command scripts can also be invoked using the IJT START command with the MEMBER(xxx) option.

The command script to start Unicenter TCPaccess Telnet Server is member START xx in the PARM data set. The command data set is specified by DDNAME SYSPROC in the runtime JCL. Member START xx is specified by CMND=START xx in the PARM field on the EXEC statement. Member START xx was established during installation and customization as described in the *Customization Guide*. Alternate START xx members can be created and overridden on the PROC CMND field. That is:

```
S RUNTLN,CMND=START99
```

Sample Command Script

The following is a sample command script to start GTF and turn tracing on for some events in a task group named TGI, and then to invoke another command script named TGICMNDs:

```
SET GTF OFF                /* TURN GTF MODE OFF FOR ALL TASK GROUPS */
SET GTF ON TG(TGI)         /* TURN GTF MODE ON FOR TGI */
MOD GTF OFF ALL            /* TURN OFF ALL GTF EVENTS */ MOD
GTF ON CB(SDWA PARM MODI) /* TURN ON DESIRED EVENTS */
%TGICMNDs                  /* INVOKE COMMAND SCRIPT FOR TGI TASK GROUP */
```

Special command statements valid only within a command script are provided to perform these tasks:

- Control the display of command statements before being executed
- Control command statement processing if a command fails

The special commands are as follows:

FLUSH | NOFLUSH

FLUSH	Specifies that the command input stack be purged (flushed) when execution of a command statement fails. This is useful to suppress further command execution if a critical command in a sequence fails.
NOFLUSH	Specifies that the command input stack continue to be processed even if execution of a command statement fails.

Default is NOFLUSH.

LIST | NOLIST

LIST	Specifies that command statements should be displayed before execution.
NOLIST	Specifies that command statements not display.

Default is LIST.

Notes

- The SRC should not be specified.
- A complete command statement must be contained in one input source record.
- Comment statements can be included anywhere. Start a comment statement with an * in column one or by enclosing the comment in /* and */ (for example: /* *comment* */).
- Sequence numbers, if included in source input, are assumed to be in the last eight columns for fixed-length records and in the first eight columns for variable-length records. Otherwise, the entire record is assumed to contain text.

Using the System Management Facility (SMF)

This chapter provides samples of SMF record formats and discusses the following topics:

- [Standard MVS SMF Record Header](#) – Sample of the standard MVS record header
- [Standard](#) – Tables for descriptor, product, task identification, and user identification sections
- [Server Application Event Records](#) – Tables for descriptor, common data, and Subtype 23 data sections
- [Interval Records](#) – Tables describing Virtual Storage Statistics (Subtype 80)
- [SMF Report Writer Program](#) – Example of JCL to generate reports for various SMF records written by Unicenter TCPaccess Telnet Server

Unicenter TCPaccess Telnet Server generates SMF records in accordance with the record formats provided in IBM publication *GC28-1030*. The Unicenter TCPaccess Telnet Server SMF record is variable length but begins with the MVS SMF record header and the Unicenter TCPaccess Telnet Server SMF record header. The Unicenter TCPaccess Telnet Server SMF record header is the anchor block used to determine which other sections are present.

On the Unicenter TCPaccess Telnet Server software tape, there is a sample of the JCL to dump Unicenter TCPaccess Telnet Server SMF records from dump data sets. In the MAC library, the SMFRECS macro maps the SMF records.

Unicenter TCPaccess Telnet Server SMF recording is specified by the SMF statement in member IJTFCGxx which is described in the *Customization Guide*.

Standard MVS SMF Record Header

The following table is a sample of the standard MVS record header.

Offset	Field	Length	Format	Source	Description	
0	0	SMFACLEN	2	Binary	Internal	Record length
2	2	SMFACSEG	2	Binary	Internal	Segment descriptor
4	4	SMFACFLG	1	Binary	SVC 83	Header flag byte
5	5	SMFACRTY	1	Binary	Internal	Record type
6	6	SMFACTME	4	Binary	SVC 83	Time, in hundredths of a second, that the record was moved to the SMF buffer
10	A	SMFACDTE	4	Packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDF, where F is the sign byte
14	E	SMFACSID	4	EBCDIC	SMCASID	System ID (taken from the SID parameter)

Standard SMF Record Header

This section provides samples of standard Unicenter TCPaccess Telnet Server SMF record headers. The record headers include Descriptor and Product sections.

Descriptor

The following table contains the descriptor section of the SMF header.

Offset	Field	Length	Format	Source	Description
18 12	SMFACNOD	2	Binary	Internal	Number of descriptors in this section
20 14	SMFACPOF	4	Binary	Internal	Offset to product section
24 18	SMFACPLN	2	Binary	Internal	Length of product section
26 1A	SMFACPNO	2	Binary	Internal	Number of product sections
28 1C	SMFACTOF	4	Binary	Internal	Offset to task information section
32 20	SMFACTLN	2	Binary	Internal	Length of task information section
34 22	SMFACTNO	2	Binary	Internal	Number of task information sections
36 24	SMFACUOF	4	Binary	Internal	Offset to user identification section

Offset	Field	Length	Format	Source	Description
40 28	SMFACULN	2	Binary	Internal	Length of user identification section
42 2A	SMFACUNO	2	Binary	Internal	Number of user identification sections
44 2C	SMFACDOF	4	Binary	Internal	Offset to data section
48 30	SMFACDLN	2	Binary	Internal	Length of data section
50 32	SMFACDNO	2	Binary	Internal	Number of data sections

Product

The following table contains the product section of the SMF header.

Offset	Field	Length	Format	Source	Description
0 0	SMFACTYP	2	Binary	Internal	Subtype identification
2 2	SMFACRVN	2	EBCDIC	Internal	Record version number - 02
4 4	SMFACNAM	8	EBCDIC	Internal	Product name
12 C	SMFACVER	2	EBCDIC	Internal	Unicenter TCPaccess Telnet Server version number
14 E	SMFACACB	8	EBCDIC	APPCFGxx	Unicenter TCPaccess Telnet Server ACB name
22 16	SMFACJBN	8	EBCDIC	Internal	Jobname of Unicenter TCPaccess Telnet Server job
30 1E	SMFACJBI	8	EBCDIC	Internal	JES job ID of Unicenter TCPaccess Telnet Server job
38 26	SMFACASI	2	Binary	Internal	Address space ID of Unicenter TCPaccess Telnet Server job
40 28	SMFACSSI	4	EBCDIC	Internal	Subsystem ID of Unicenter TCPaccess Telnet Server job

Server Application Event Records

Task Identification Section

The following table contains the task identification section (pseudo-task data) of the Unicenter TCPaccess Telnet Server SMF header.

Offset	Field	Length	Format	Source	Description
00	SMFACPGM	8	EBCDIC	PTAPGM	Program name

User Identification Section

The following table contains the user identification section (AC#U data) of the Unicenter TCPaccess Telnet Server SMF header:

Offset	Field	Length	Format	Source	Description	
0	0	SMFACUID	8	EBCDIC	SECUID	RACF (or other external security system) user ID
8	8	SMFACGID	8	EBCDIC	SECGRP	RACF (or other external security system) group ID
16	10	SMFACACT	144		AC#UACCT	Accounting information

The following is the format of the SMFACACT field:

- First byte of field contains the number (binary) of accounting fields
- Following bytes contain accounting fields, where each entry for an accounting field contains a one-byte length field, followed by the field

Telnet Server Record (Subtype 23) Data Section

The SMF subtype 23 record is written by the telnet server upon termination of a session with a VTAM session partner. Several subtype 23 records may be written for a single telnet session. The subtype 23 record contains a User and a Task section.

Offset	Length	Label	Format	Source	Description
00		S23DATA			Data section
00	04	S23LIPA	Binary	TCCLINET	Local host IP address
04	04	S23RIPA	Binary	TCCRINET	Remote host IP address
08	02	S23LPOR	Binary	TCCLINET	Local host port
0A	02	S23RPOR	Binary	TCCRINET	Remote host port
0C	08	S23SNET	EBCDIC		Netid associated with SLU (TN3270E server only)
14	08	S23SLU	EBCDIC	LURQLUN	ACB name (SLU)
1C	08	S23PNET	EBCDIC		Netid associated with PLU (TN3270E server only)
24	08	S23PLU	EBCDIC	VCBPART	PLU name
	81				Length of Subtype 23 data section
2C	04	S23STIME	Binary	TIME macro	Session start time, in hundredths of a second since midnight
30	04	S23SDATE	Binary	TIME macro	Session start date, in form 0cyydddF, where: c = century indicator (0=19, 1=20) yy = decimal year within century ddd = decimal Julian day
34	04	S23TTIME	Binary	TIME macro	Session termination time
38	04	S23TDATE	Binary	TIME macro	Session termination date
3C	04	S23DURA	Binary		Session duration, in hundredths of a second
40	04	S23CLIN	Binary	internal	Number of bytes received from Telnet client
44	04	S23CLOU	Binary	internal	Number of bytes transmitted to Telnet client
48	04	S23VTIN	Binary	internal	Number of bytes received from VTAM application
4C	04	S23VTOU	Binary	internal	Number of bytes sent to VTAM application
50	01	S23TTYPL	Binary	TCTTYPEL	Length of terminal type field
51	40	S23TTYPE	EBCDIC	TCCTTYPE	Terminal type negotiated with Telnet client
79	08	S23LMODE	EBCDIC	VCBPART	Logmode
	81				Length of Subtype 23 data section
The following fields are present for T01S3270 (TN3270E server records only)					
81	4	S23SESS	Binary		TCP connection number

Offset	Length	Label	Format	Source	Description
85	8	S23APPL	EBCDIC		Requested application name
8D	4	S23VBK1	Binary		VTAM Bucket 1 = time < 1 second
91	4	S23VBK2	Binary		VTAM Bucket 2 = time < 2 second
95	4	S23VBK3	Binary		VTAM Bucket 3 = time < 5 second
99	4	S23VBK4	Binary		VTAM Bucket 4 = time < 10 second
9D	4	S23VBK5	Binary		VTAM Bucket 5 = time >= 1 second
A1	4	S23TBK1	Binary		TCP Bucket 1 = time < 1 second
A5	4	S23TBK2	Binary		TCP Bucket 2 = time < 2 second
A9	4	S23TBK3	Binary		TCP Bucket 3 = time < 5 second
AD	4	S23TBK4	Binary		TCP Bucket 4 = time < 10 second
B1	4	S23TBK5	Binary		TCP Bucket 5 = time >= 1 second

Access Method and Protocol Layer Event Records

Interval Records

The periodic, interval-driven records are written every so often, based on the INTERVAL parameter of the SMF statement in member IJTFCGxx. The interval-driven records include subtype 80. The INTERVAL parameter is described in the *Customization Guide*.

Virtual Storage Statistics (Subtype 80) Data Section

The SMF record subtype 80 (VMSTATS) is written periodically (governed by the INTERVAL parameter of the SMF statement in the IJTFCGxx configuration member) to record virtual storage manager statistics for subpools. It contains no Task or User Identification sections.

Offset	Length	Label	Format	Source	Description
00	04	S80SPOFF	Binary	Internal	Offset to subpool descriptor section
04	04	S80SPNUM	Binary	Internal	Number of subpool descriptor entries (256)
08	04	S80IPOFF	Binary	Internal	Offset to IFS pool descriptor section

Offset	Length	Label	Format	Source	Description
OC	04	S80IPNUM	Binary	Internal	Number of IFS pool descriptor entries
10	04	S80PVT	Binary	VSMREGN	Address of start of private area region (<16m)
14	04	S80PVTSZ	Binary	VSMREGN	Size of private area region (<16m)
18	04	S80EPVT	Binary	VSMREGN	Address of start of extended private area region
1C	04	S80EPVTS	Binary	VSMREGN	Size of extended private area region
20	04	S80ALLO	Binary	VSMLIST	Total allocated in all private area subpools (<16m)
24	04	S80FREE	Binary	VSMLIST	Total free in all private area subpools (<16m)
28	04	S80USED	Binary	VSMLIST	Total used in all private area subpools (<16m)
2C	04	S80EALLO	Binary	VSMLIST	Total allocated in all private area subpools in extended region
30	04	S80EFREE	Binary	VSMLIST	Total free in all private area subpools in extended region
34	04	S80EUSED	Binary	VSMLIST	Total used in all private area subpools in extended region
38	04	S80TALLO	Binary	VSMLIST	Total allocated in all private area subpools
3C	04	S80TFREE	Binary	VSMLIST	Total free in all private area subpools
40	04	S80TUSED	Binary	VSMLIST	Total used in all private area subpools
44	04	S80RGSZ	Binary	VSMLIST	Total region size (<16m)
48	04	S80RGUN	Binary	VSMLIST	Total unallocated space (<16m)
4C	04	S80RGUS	Binary	VSMLIST	Total region used (<16m)
50	04	S80ERGSZ	Binary	VSMLIST	Total region size (>16m)
54	04	S80ERGUN	Binary	VSMLIST	Total unallocated space (>16m)
58	04	S80ERGUS	Binary	VSMLIST	Total region used (>16m)
5C	04	S80TRGSZ	Binary	VSMLIST	Total region size
60	04	S80TRGUN	Binary	VSMLIST	Total unallocated space
64	04	S80TRGUS	Binary	VSMLIST	Total region used
68	2868	S80SPLS	Binary	VSMLIST	Subpool descriptor area (described below)
2860	Varies	S80IFSPL	Binary	IPHD	IFS Pool descriptor section
	2800	S80LEN	Binary		Minimum length of subtype 80 record

Detailed Description of S80SPLS Area

The following table is a detailed description of the S80SPLS area described in the previous table, [Virtual Storage Statistics \(Subtype 80\) Data Section](#). This area is repeated 256 times (once for each subpool):

Offset	Length	Label	Format	Source	Description
00	02	S80SPID	EBCDIC	VSMLIST	Subpool id
02	02				Reserved
04	04	S80SPALO	Binary	VSMLIST	Total allocated in subpool (<16m)
08	04	S80SPFRE	Binary	VSMLIST	Total free in subpool (<16m)
0C	04	S80SPUSE	Binary	VSMLIST	Total used in subpool (<16m)
10	04	S80SPEAL	Binary	VSMLIST	Total allocated in subpool in extended region
14	04	S80SPEFR	Binary	VSMLIST	Total free in subpool in extended region
18	04	S80SPEUS	Binary	VSMLIST	Total used in subpool in extended region
1C	04	S80SPTAL	Binary	VSMLIST	Total allocated in subpool
20	04	S80SPTFR	Binary	VSMLIST	Total free in subpool
24	04	S80SPTUS	Binary	VSMLIST	Total used in subpool
	28	S80SPLEN			Length of subpool descriptor section

The following table is a detailed description of the IFS Pool descriptor section (S80IFSPL) described in the [Virtual Storage Statistics \(Subtype 80\) Data Section](#).

Detailed Description of S80IFSPL Area

Offset	Length	Label	Format	Source	Description
00	04	S80IPID	EBCDIC	IPHD	IFS pool ID
04	01	S80IPSP	Binary	IPHD	Subpool
05	03	S80IPLN	Binary	IPHD	Length of elements
08	01	S80IPFLG	Binary	IPHD	Flags: X'80' S80IPFSP Requests may be suspended X'40' S80IPFFX Permanently fixed storage X'10' S80IPFHD Headers separate X'08' S80IPFEX Format exit exists
09	03		Binary		Reserved

Offset	Length	Label	Format	Source	Description
0C	04	S80IPTTL	Binary	IPHD	Total elements in pool
10	04	S80IPFRE	Binary	IPHD	Count of free entries
14	04	S80IPFLW	Binary	IPHD	Low-water mark of free entries
18	04	S80IPHWM	Binary	IPHD	Highest count of entries
1C	04	S80IPREQ	Binary	IPHD	Number of requests from pool
20	04	S80IPEXP	Binary	IPHD	Times expanded
24	04	S80IPCNT	Binary	IPHD	Times contracted
28	04	S80IPWCT	Binary	IPHD	Number of callers who waited
2C	04	S80IPERR	Binary	IPHD	Number of callers in error
	30	S80IPLN	Binary		Length of IFS pool descriptor section

SMF Report Writer Program

The SMF Report Writer Program, T00SMFRW, generates several types of reports for the various SMF records written by Unicenter TCPaccess Telnet Server.

To execute the Report Writer, use the following JCL:

```
//SMFRW      EXEC  PGM=T00SMFRW
//STEPLIB    DD   DISP=SHR,DSN=TRGIND X.LINK
//SMFIN      DD   DISP=SHR,DSN=SMFDATA
//SYSPRINT   DD   SYSOUT=*
//SYSIN      DD   *
/*
```

Where:

SMFIN DD	Defines the file holding the SMF records. This file can be created from the system SMF data sets using the IBM IFASMFDP utility.
SYSPRINT DD	Defines the file where the report will be written.
SYSIN DD	Defines the file that holds the Report Writer commands.

Commands

The following commands are recognized by the report writer:

RECORDTYPE=*record_type*

Defines the SMF record type used by Unicenter TCPaccess Telnet Server.

Valid types are 128-255.

Default: 130.

DATE=[*date* | (*start_date-end_date*)]

Defines the date or range of dates for which SMF records will be selected for the report.

Format: *yyyyddd*, where

yyyy = year

ddd = Julian day of year

Default: All dates.

TIME=[*time* | (*start_time-end_time*)]

Defines the time or range of times for which SMF records will be selected for the report.

Format:*hh:mm:ss.th*, where:

hh = the hour

mm = minutes

ss = seconds

th = tenths and hundredths of a second.

Any precision not specified is assumed to be zero.

For instance, 12:10 = 12:10:00.00.

Default: All times

REPORT= <i>report_type</i>	Defines the report type to be written. Several REPORT commands can be included to combine reports. The report types are described in the following table.
LOCAL= <i>ip_address:port</i>	Limits the report to records matching the local IP address and port specified.
REMOTE= <i>ip_address:port</i>	Limits the report to records matching

	the remote IP address and port specified.
JOBNAME= <i>jobname</i>	Limits the report by the jobname of the Unicenter TCPaccess Telnet Server job.

The report types in the following table can be specified using the REPORT= command.

Report Type	Description	Record Subtypes
VSMSTATS	VSM statistics (totals)	80
VSMPOOLS	VSM Statistics by subpool	80
IFSPOLS	IFS pool activity	80
TELNET	Telnet activity	23
DUMP	Unformatted dump of records	All
ALL	All reports (except DUMP)	All

Note:

- The DUMP report should not be combined with another report type
- The LOCAL and REMOTE commands apply to the Telnet reports

The IP address or port can be specified alone.

Example

```
LOCAL=138.42.220.120:1223
REMOTE=138.42.220.85
LOCAL=:23
```


Diagnostic Procedures

This chapter describes the problem determination and reporting procedures for Unicenter TCPaccess Telnet Server.

This chapter discusses the following topics:

- [Problem Determination](#) – Describes problem diagnostic procedures and how to use the diagnostic commands
- [Problem Reporting](#) – Describes the documentation needed when reporting problems to customer support

Problem Determination

If Unicenter TCPaccess Telnet Server does not respond properly, diagnostic action is required to restore proper service. This section provides a background in performing diagnostic action and gathering documentation for problem resolution.

Although there can be many different types of software failures, the majority of them fit into some general categories.

Whenever a problem is suspected, a good first step in problem determination is to check the T01LOG and JES message log (or console log) for any messages that may indicate the nature of the problem. You can use the IFS LOGGING command to enable more diagnostic messages, such as debugging messages. For more information, see [LOGGING](#).

Failures That Produce Dumps

In the event of an abend, a system dump (SVC dump) is produced. This is usually accompanied by message T00IF050E in the message log.

- Save the dump/logs
- Complete Unicenter TCPaccess Telnet Server job output
- Forward to Customer Support

For certain abends, the dump is suppressed—for instance, if an external application passes an invalid buffer address. In this case, an abend message may appear in the log without an accompanying dump. If this happens frequently, the application should be identified and corrected.

Failures Due To Network Problems

If the problem is reproducible, use the realtime network trace facility, TCPEEP, to diagnose the problem. For intermittent problems, use the TRACE CT command to initiate a non-wrapping trace of the network interface(s) involved. For more information, see [TCPEEP](#) and [TRACE](#).

Failure of Unicenter TCPaccess Telnet Server to Respond

Certain problems can lead to a failure of Unicenter TCPaccess Telnet Server, or a major application, to respond. Try to identify the failing component.

Failure of Major Applications

You can use the TSO NETSTAT and SYSSTAT commands or the operator console to query and cancel connections. For more information, see [NETSTAT/SYSSTAT](#).

Loops

Unicenter TCPaccess Telnet Server has loop detection code in most cases, to cancel a process that is looping, and take a diagnostic dump. If Unicenter TCPaccess Telnet Server is not responding, and using large amounts of CPU, it may be necessary to cancel the job. First, take a dump of the address space using the Unicenter TCPaccess Telnet Server SVCDUMP command, or, if that fails to respond, use the console DUMP command.

Storage shortages

If the amount of virtual storage used by Unicenter TCPaccess Telnet Server exceeds the threshold defined on the MAXSTGPCT parameter of the IFSPARM statement in the IJTCTGxx configuration member, it stops accepting new connections. This condition may also be accompanied by message T00IF092W on the console.

Use the IFS POOL command and VSM command to display virtual storage allocation. Use the IFS SVCDUMP command to take a diagnostic dump of the Unicenter TCPaccess Telnet Server address space. In some cases, it may be possible to use NETSTAT to free up storage by canceling old or inactive sessions. For more information, see NETSTAT/SYSSTAT in the chapter “Diagnostic Commands”.

If the problem is chronic and involves a slow leak of virtual storage, use the SMF statement of the IJTCTGxx configuration member to turn on INTERVAL recording and record subtype 80. Be sure your system SMF parameters and exits (such as IEFU85) allow the recording of the Unicenter TCPaccess Telnet Server SMF record type. When the SMF data is collected, you can use the SMF Report Writer Program to view storage utilization and growth during the recording period. For more information, see SMF Report Writer Program in the chapter “Using the System Management Facility (SMF).”

Note: Use of the product ABEND-Aid is not recommended for Unicenter TCPaccess Telnet Server. Dumps formatted by this product do not include many of the control blocks required for problem analysis.

Problem Reporting

When reporting a Unicenter TCPaccess Telnet Server problem to customer support, document as much information as you can to fully characterize the state of the system at the time of failure and any detectable failure symptoms. The exact documentation collected for a given problem varies, depending on the type of failure experienced.

In general, the following items should always be provided as initial documentation of a failure:

- The release and maintenance level of the product and function
- Always include T01LOG output
- The console log, or the job log, if running in batch
- The conditions leading up to the failure
- Whether the failure is reproducible or a one-time occurrence

- Whether this is a new failure or if the product was working until this happened (maintenance, more users, new application)

It is better to collect an abundance of initial documentation than wait to have Technical Support technicians request additional information later when the data may not be available (especially in the case of intermittent failures or hardware problems).

ABENDs

In the event of an ABEND, collect the following information:

- The SVC dump of the failing address space.
- A detailed description of the action immediately preceding the failure (if known) or conditions of the system (for example, a large number of Telnet sessions).
- The job output of the Telnet Server at the time of failure, including the JES log, T01LOG, and any enabled traces.
- Whether the failure is reproducible or random. If reproducible, obtain a TCPEEP trace while recreating the failure. If random, you may be able to capture the trace data using the TRACE command to start a wrap-around trace. See [TCPEEP](#) and [TRACE](#) for more information.

Incorrect Output

For incorrect output through Telnet, collect the following information:

- A detailed description of the action immediately preceding the failure (if known) or conditions of the system (for example, a large number of Telnet sessions).
- The job output of the Telnet Server at the time of failure, including the JES log, T01LOG, and any enabled traces.
- Any client software involved, including the name of the vendor and the release number.
- Whether the failure is reproducible or random. If reproducible, obtain a TCPEEP trace and a VTAM buffer trace while recreating the failure.

Startup and Parameter Errors

For startup problems and parameter errors, collect the following information:

- The job output of the Telnet Server at the time of failure, including the JES log, T01LOG, and any enabled traces.

- A listing of the parameter and command members affected (especially any recent changes).
- Whether the failure is reproducible or random.

Diagnostic Commands

This chapter describes the diagnostic commands available in Unicenter TCPaccess Telnet Server.

It includes the following topics:

- [STROBE](#) – A command to measure the amount of CPU time used by a module
- [TCPEEP](#) – A command to invoke the Unicenter TCPaccess Telnet Server packet trace program and diagnose remote host communication problems
- [TRACE](#) – Trace command enhancements to collect TCP/IP data and display it on a terminal or send it to an external writer

STROBE

STROBE is a diagnostic tool to measure the amount of CPU time, by module. The tool is first activated using a STROBE ON command, time sampling is started, and at each time slice, the address portion of the interrupted PSW is saved. When the desired amount of samples is reached, or when the run is forced with the STROBE OFF command, module names are identified by matching the captured address list against module residence addresses.

The subsequent report is sorted by popularity, in decreasing order, and makes a distinction between TCB mode vs. SRB mode hits.

NOTE: The report is produced when the STROBE process ends (because the sample limit was reached, or because a STROBE OFF command is issued), and is sent to the T01LOG as diagnostic-level messages.

Syntax for the STROBE command:

```
STROBE ON|OFF|STATUS <SAMPLES(nnn) DETAIL(name)>
```

ON	Activates the sampling run. Sampling runs until the SAMPLES count is reached, or stopped with the OFF operand.
OFF	Used to terminate the run when desired.

STATUS	Used at anytime to sample progress.
SAMPLES(<i>nnn</i>)	Specifies the number of samples to capture. The upper limit is 1000000. May be aliased as SIZE. Default: 500000.
DETAIL(<i>name</i>)	Used to focus on a specific module for more detailed information.

TCPEEP

TCPEEP is a TSO command that invokes the Unicenter TCPaccess Telnet Server packet trace program to diagnose remote host communication problems. TCPEEP creates a NO WRAP Component Trace Instance and displays the output on a TSO terminal or directs it to a dynamically allocated SYSOUT data set. Optionally, it can stop any component trace instance, or modify an existing component trace instance or view an existing Component Trace Instance.

Note: TCPEEP runs only when the Unicenter TCPaccess Telnet Server and TRACE address spaces are active.

User Interface

TCPEEP is run as a TSO command, either online or in batch.

The JCL to run TCPEEP as a batch job is in SAMP member TCPEEP. Unicenter TCPaccess Telnet Server TRACE must be up and running before submitting a batch job for TCPEEP.

The following is sample JCL for running TCPEEP in batch.

```
//TCPEEP JOB (TCPEEP), 'TCPEEP', CLASS=A, MSGCLASS=X
//*
//* Sample JCL to run TCPEEP in batch.
//*
//* Update 'trgindx' to reflect your library naming convention.
//*
//* Note: The TCPaccess and TRACE address spaces must be
//* running.
//*
//TCPEEP EXEC PGM=IKJEFT01, DYNAMNBR=50, REGION=4M
//STEPLIB DD DISP=SHR, DSN=trgindx.LINK
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*, LRECL=132, BLKSIZE=132, RECFM=FB
//SYSIN DD DUMMY
//*
//SYSTSIN DD *
TCPEEP TRCSSID(ACR) GROUPS(TELNET) +
      BUFFTME(5)
```

```
/*
```

To stop the TCPEEP batch job, issue the MVS **STOP** command (for example, **P jobname**).

Viewing Trace Data

TCPEEP collects realtime data and by default writes to SYSTSPRT. Optionally, it can write to a dynamically allocated SYSOUT data set.

Trace Operation

To stop a trace, press the terminal ATTENTION key and enter **H** at the prompt (null entry allows trace to continue).

WARNING! *TCPEEP should be installed in a protected library only. It can be used to display all network traffic through the Telnet Server, including user IDs and passwords.*

TCPEEP Syntax

This section describes the TCPEEP syntax and describes its parameters.

```
TCPEEP [ ASID( asid,... ) ]
      [ BUFFERS ( size, number ) ]
      [ BUFFTIME ( time_out ) ]
      [ DATASIZE( record_size ) ]
      [ FORMAT( format_options ) ]
      [ FULL | SUMMARY ]
      [ GROUPS( ( group [, 'filter' ) ... ) ]
      [ HALT ]
      [ INSTANCE ( instance_ID ) ]
      [ JOBNAME ( jobname,... ) ]
      [ NOHEADER ]
      [ PEEK ( limit ) ]
      [ SYSOUT ( class ) ]
      [ TRACESIZE ( num_records ) ]
      [ TRCSSID ( ssid ) ]
```

ASID (*asid* ,...)

Specifies the address space identifiers (ASIDs) of address spaces used as a filter for tracing. Events in the ASIDs are recorded by the component trace.

The parameter contains a list of 0 to 16 hexadecimal ASIDs separated by commas.

An empty ASID list, ASID=(), turns off filtering by address spaces. In the ASID parameter, list all address spaces to trace. Address spaces for previous traces are not traced unless listed.

Default: None.

BUFFERS (*size,number*)

Specifies the size of the trace buffers in kilobytes or the number of buffers.

size A value between 64 and 1024 (Default: 256).

num A value between 2 and 128 (Default: four).

BUFFERS (Optional). Can only be specified when creating a new trace instance. If specified when modifying a trace instance, it is ignored.

Range: *Size*: 64 – 1024
 Number: 2 – 128.

Default: (256, 4).

BUFFTIME (*time_out*) Specifies the buffer time out interval in seconds. At the end of each interval, if the current buffer contains data but is not full, a buffer flush operation is initiated.

BUFFTIME is optional and can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

Use this parameter to force a buffer switch so you do not have to wait for the entire buffer to fill to see trace data.

Range: 0 – 99999.

Default: 10.

DATASIZE (*record_size*)

Specifies the maximum size of a trace record in kilobytes. Trace records that exceed the specified value are truncated.

DATASIZE – (Optional). Can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

If the specified maximum size exceeds the largest supported trace record size (64 KB less control headers), then the specification has no effect.

Default: None.

FORMAT (format_options)

The format of the collected data.

Formatted data can be specified as follows:

TNDATA | TNEBCDIC | TNASCII

or

TNDATA(*nn*) | TNEBCDIC(*nn*) | TNASCII(*nn*)

Displays details about TN3270 parameters and optionally data

- *nn* is between 0 and 65535 displays data captured for the trace group TELNET
- TNDATA uses an ASCII translate table for NVT mode and EBCDIC otherwise
- TNEBCDIC uses an EBCDIC-only translate table
- TNASCII uses an ASCII-only translate table

Note: TNDATA is recommended, since it translates data according to the session's current negotiated telnet protocol.

FULL | SUMMARY Show all or a summary of the data.

Default: SUMMARY.

GROUPS((*group* [, '*filter*']) ...)

Trace group or groups for which data is collected and optionally a filter parameter for each group.

Use the GROUPS parameter to limit the amount of data collected. Used with INSTANCE to modify an existing trace instance, either adds a new group to the trace instance or replaces an existing group for the trace instance. Once added, a group cannot be removed from the active trace instance.

You can specify a maximum of four trace groups.

group

Selects the type of data to collect: RTM, or TELNET.

- TELNET is a collection of trace points in the TN3270E server to trace activity to/from the client and the VTAM application.

HOST(*host,...,host*) - up to 16 IP HOST addresses names).

PORT(*port,...,port*) - up to 16 port numbers (names).

LU(*luname,...,luname*) - up to 16 LU names. These are either VTAM Applids or SLU names. This filter is restrictive in that, if specified, no tracing should be collected on a connection until either an SLU or APPLID is determined.

MAXTNDATA(*nn*). *nn* is between 1 and 65535. By default, no data is captured. (Alias is MTNDATA).

Refer to [Useful TCPEEP Commands](#) for usage of the GROUPS parameter.

- RTM - is a collection of trace points in TelnetRTM that traces RTM activity. The following filter options are supported:
 - UASID(*asid,...,asid*) - takes 1 to 16 *asids*, which refer to jobs using TelnetRTM (Unicenter NetSpy or a Unicenter TCPaccess Telnet Server).
 - USER(*jobname,...,jobname*) - takes 1 to 16 jobnames, which refer to jobs using TelnetRTM (Unicenter NetSpy or a S Telnet server).

Default: NETIF.

HALT

Stops a component trace instance.

INSTANCE (*instance_id*)

Select a trace instance to display by specifying the *instance_id* returned from TCPEEP or the MVS TRACE CT command.

JOBNAME (*jobname ,....*)

Names of jobs used as filters for tracing. Events in these jobs are recorded by the component trace.

NOHEADER	<p>Do not display the header information from the trace entry.</p> <p>This option helps limit the output. Without it, one line prints describing the trace entry even if no other information about that entry displays.</p>
PEEK (<i>limit</i>)	<p>Number of trace records to view.</p> <p>Use to view an existing trace. PEEK or PEEK(0) implies no limit; PEEK(<i>n</i>) traces only <i>n</i> trace records.</p> <p>Range: Zero - no limit.</p> <p>Default: Zero.</p>
SYSOUT (<i>class</i>)	<p>Sends output to a dynamically allocated SYSOUT data set.</p> <p>By default, the output for TCPEEP writes to SYSTSPRT.</p> <p>Default: X.</p>
TRACESIZE (<i>num_records</i>)	<p>Maximum number of trace records to be recorded. If not specified, there is no limit to the number of records recorded.</p> <p>TRACESIZE is optional and can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.</p>
TRCSSID (<i>ssid</i>)	<p>ID of the trace address space.</p> <p>Default: ACTR.</p>

Useful TCPEEP Commands

This section lists the keywords for each function listed.

Creating a NOWRAP Trace	<p>Use the following command to initiate a NO WRAP trace instance:</p> <pre>TCPEEP TRCSSID(<i>trcsubsysid</i>) GROUPS(<i>group</i>) + JOBNAME(<i>jobname</i>) ASID(<i>asid</i>) DATASIZE(<i>size</i>) TRACESIZE(<i>trcsize</i>) + BUFFERS(<i>bufname</i>) BUFTIME(<i>buftime</i>) FORMAT(<i>format_options</i>) + SYSOUT(<i>class</i>)</pre>
Stopping Trace	<p>Use this command to stop any trace instance:</p> <pre>TCPEEP INSTANCE(<i>inst</i>) HALT TRCSSID(<i>trcsubsysid</i>)</pre>
Modify Existing Trace	<p>Use this command to modify an existing trace instance:</p> <pre>TCPEEP INSTANCE(<i>inst</i>) TRCSSID(<i>trcsubsysid</i>) +</pre>

```
GROUPS ( group ) JOBNAME( jobname ) ASID( asid )      PEEK          +  
FORMAT( format_options ) SYSOUT( class )
```

Regardless of how the trace is started view its LIFO for the specified number of records (PEEK or PEEK(0) implies no limit).

View Existing Trace

Use this command to view an existing trace instance:

```
TCPEEP INSTANCE( inst ) TRCSSID( trcssubsysid ) PEEK      +  
FORMAT( format_options ) SYSOUT
```

Regardless of how the trace is started view its LIFO for the specified number of records (PEEK or PEEK(0) implies no limit).

Specifying GROUPS

This example shows how to specify the GROUPS for TCPEEP.

```
TCPEEP GROUPS ( ( TELNET, 'HOST ( ip1, ip2 ), PORT ( 23 ) ' ),      +  
                ( TELNET, 'HOST ( ip3 ) ' ) )
```

The following example allows you to specify NETIF without additional filtering.

```
TCPEEP GROUPS ( TELNET )
```

TCPEEP Example

TELNET Trace

This sample shows the output for a TELNET command.

```
TNRECV  000A0303 05/07 14:23:35.312784 Telnet - TN Recv  
T4SB: Sess=3, Appl=TSO      , SLU=*****  
Local=141.202.198.131:3203, Remote=130.200.174.43:2214  
DATA +0000 FFFD19          *...          *  
-----  
TNINPUT 000A0301 05/07 14:23:35.312839 Telnet - TN Input  
TN: Length=00000001 (1), Hdr=0000000000, Pgm=T04STNEG  
T4SB: Sess=3, Appl=TSO      , SLU=***** , State=NU , NegState=DR  
Local=141.202.198.131:3203, Remote=130.200.174.43:2214  
<-- IAC DO END-OF-RECORD  
DATA +0000 19              * .          *  
-----  
VNOPEN  000A0101 05/07 14:23:35.313044 Telnet - VN Open  
VN: PLU=      , func=LU_Get , LU=A03VLT20, Pool=LUPPOOL  
T4SB: Sess=3, Appl=TSO      , SLU=*****  
-----  
VNOPEN  000A0101 05/07 14:23:35.314874 Telnet - VN Open  
VN: PLU=      , func=ACB-OPEN, PCID=0000000000000000  
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20  
-----  
VNSESSI 000A0103 05/07 14:23:35.315103 Telnet - VN Initiate session  
VN: PLU=TSO      , func=SETLOGON, PCID=0000000000000000  
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20  
-----  
VNSESSI 000A0103 05/07 14:23:35.315323 Telnet - VN Initiate session  
VN: PLU=TSO      , func=X-SETLGN, PCID=0000000000000000  
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20  
-----  
VNSESSI 000A0103 05/07 14:23:35.316338 Telnet - VN Initiate session
```

```

VN: PLU=TSO      , func=REQSESS , PCID=0000000000000000
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
-----
VNSESSI 000A0103 05/07 14:23:35.317801 Telnet - VN Initiate session
VN: PLU=      , func=X-REQ , PCID=0000000000000000
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
-----
VNXITS 000A0109 05/07 14:23:35.386844 Telnet - VN SCIP, NSEXIT, TPEND
VN: PLU=A71IT033, func=BIND , LOGMODE=LSX32704,
PCID=DEFF541356DBDFB2
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
T4VS: LU2, State=WAITBIND
-----
VNSESSI 000A0103 05/07 14:23:35.386932 Telnet - VN Initiate session
VN: PLU=A71IT033, func=OPNSEC , PCID=DEFF541356DBDFB2
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
-----
VNSESSI 000A0103 05/07 14:23:35.389260 Telnet - VN Initiate session
VN: PLU=A71IT033, func=X-OPNSEC, PCID=DEFF541356DBDFB2
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
-----
VNSESSI 000A0103 05/07 14:23:35.391649 Telnet - VN Initiate session
VN: PLU=A71IT033, func=SDT , PCID=DEFF541356DBDFB2
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
-----
VNSESSI 000A0103 05/07 14:23:35.391784 Telnet - VN Initiate session
VN: PLU=A71IT033, func=X-SDT , PCID=DEFF541356DBDFB2
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
-----
VNSESSI 000A0103 05/07 14:23:35.396801 Telnet - VN Initiate session
VN: PLU=A71IT033, func=StrtSess, PCID=DEFF541356DBDFB2
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
-----
VNRCV0 000A0107 05/07 14:23:35.396947 Telnet - VN Receive started
VN: PLU=A71IT033, func=RCV0 , PCID=DEFF541356DBDFB2
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
-----
VNRECV 000A0108 05/07 14:23:35.397082 Telnet - VN Receive completed
VN: PLU=A71IT033, func=X-RCV0 , PCID=DEFF541356DBDFB2
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
T4RB: Receive: DATA Length=0000002F (47) BB:NEB OIC NCMD DEF
Seq=0001:0000
-----
VNRECV 000A0108 05/07 14:23:35.401066 Telnet - VN Receive completed
VN: PLU=A71IT033, func=RecvData, PCID=DEFF541356DBDFB2
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20
-----
TNSEND 000A0302 05/07 14:23:35.401096 Telnet - TN Send
TN: Length=00000031 (49)
T4SB: Sess=3, Appl=TSO      , SLU=A03VLT20, State=VO , NegState=NC
Local=141.202.198.131:3203, Remote=130.200.174.43:2214
DATA +0000 7EC111F5 6F1D4011 F56F1D40 1140401D *=A.5?. .5?. . .*
+0010 C8C9D2D1 F5F6F7F0 F0C140C5 D5E3C5D9 *HIKJ56700A ENTER*
+0020 40E4E2C5 D9C9C440 601D4011 C15013FF * USERID -. .A&..*
+0030 EF * *

```

TRACE

Unicenter TCPAccess Telnet Server contains enhancements to the IBM MVS Component Trace facility to provide a method of collecting Telnet data and displaying it on a terminal or sending it to an external writer. Additional JCL is required to enable the Component Trace enhancements. To use Component Trace as a TSO command, see [TCPEEP](#).

MVS Component Trace

MVS Component Trace is a diagnostic aid used to trace the action of certain system components and third party components that define themselves to Component Trace. The TRACE operator command is used to start, stop and control the component trace. For more information on the MVS TRACE command, see the IBM MVS System Commands.

Trace Address Space

Component Trace is defined in its own address space and collects trace data for trace points defined in other address spaces. Likewise, the other address spaces must identify the Component Trace address space for the data collection. Multiple occurrences of Component Trace can be active at the same time, each with a unique subsystem ID.

For each address space, IJTFCGxx in the PARM member defines the subsystem ID. The definition is specified using the TRACENAME keyword on the IFSPARM statement as follows:

```
IFSPARM PROMPT TRACENAME( ACTR ) NOPROMPT
```

In the example, the trace data in the address space is directed to the Component Trace address space with a ACTR subsystem ID.

Note: If the Trace Address Space is brought up after the TCP stack, there may be a delay of two minutes before events are traced.

Exit

MVS Component Trace requires an exit to communicate with the tracing component. The T03PTRSS exit must reside in LPALIB or the Link List. It is distributed in the LINK library.

External Writer

The collected trace data is written to DASD or TAPE using an external writer. A suitable External Writer Cataloged Procedure for use with Component Trace may already be defined on your system. For more information on defining an External Writer used with Component Trace, see the IBM publication *MVS Diagnosis: Tools and Service Aids*.

Component Trace JCL

This JCL allows you to run a Component Trace Address Space.

WARNING! Make sure that the EXEC statement specifies an IJTFCGnn member that has **only** TRACE in the IFSPARM APPS parameter. If any other applications are given, the address space may not start, or it may start with incorrect applications activated. **The latter condition can be corrected only after an IPL.**

```
//RUNTRACE JOB
//*
//*   SAMPLE JCL PROCEDURE TO RUN TCPACCESS'S TRACE ADDRESS SPACE
//*   NOTE: THIS ADDRESS SPACE SHOULD NOT BE TERMINATED
//*
//*   EDIT THE TRGINDX, SSN, SOUT, CMND SYMBOLIC
//*   PARAMETERS
//*
//*   VERIFY THAT THE JOB CARD AND NAMING CONVENTIONS MEET
//*   YOUR SITE'S JCL REQUIREMENTS, THEN SUBMIT THIS JOB.
//*
//*   WARNING: Make sure that the EXEC statement below specifies an
//*             IJTFCGnn member that has *only* TRACE in the IFSPARM
//*             APPS parameter. If any other applications are given,
//*             the address space may not start, or it may start with
//*             incorrect applications activated. *** The latter
//*             condition can be corrected only after an IPL.***
//*
//ICSTRACE PROC TRGINDX='TRGINDX', TARGET LIBRARIES DSN INDEX
//              SSN=ACTR,           DFLT SUBSYSTEM NAME
//              SOUT='*',           CHOOSE A HOLD NONPURGE SYSOUT CLASS
//              CMND=STARTTR,       DFLT STARTUP COMMAND SCRIPT NAME
//              CNFG=TR             IJTFCGXX SUFFIX
//*
//TRACE      EXEC PGM=IFSSTART,REGION=6144K,TIME=1440,
// PARM='IFSINIT,U=&SSN,SO=&SOUT,CM=&CMND,CF=&CNFG'
//*
//STEPLIB DD DISP=SHR,DSN=&TRGINDX..LOAD
//          DD DISP=SHR,DSN=&TRGINDX..SASLINK
//*
//* WARNING: THE LOAD DATA SET MUST NEVER BE ADDED TO THE LINK LIST.
//*           IOS/390'S ELEMENT NAMES ARE NOT UNIQUE AND MAY
//*           AFFECT THE OPERATIONS OF OTHER SOFTWARE.
//*           THE LIBRARY SHOULD ALWAYS BE REFERENCED THROUGH
//*           A STEPLIB OR JOBLIB STATEMENT.
//*
//*           CONFIGURATION DATA SETS
//*
//SYSPARM DD DISP=SHR,DSN=&TRGINDX..PARM
//SYSPROC DD DISP=SHR,DSN=&TRGINDX..PARM
//SYSHELP DD DISP=SHR,DSN=&TRGINDX..HELP
//*
```

```
/**      LOG DATA SETS
/**
//T01LOG DD  SYSOUT=&SOUT
/**
/**      DUMP DATA SETS
/**
//SYSUDUMP DD  SYSOUT=&SOUT
/**
/**      MISC DATA SETS
/**
//ABNLIGNR DD  DUMMY          /* DISABLE ABEND-AID PROCESSING */
```

TRACE Command

The TRACE command starts, stops, modifies, or displays the status of a system trace, master trace, or component trace.

The TRACE command is a standard MVS operator command.

```
TRACE      [ CT { [ , ON | OFF ] [ , COMP=name ] [ , PARM=mem ] } ]
           [ CT { [ , WTRSTART=mem_name[, WRAP | NOWRAP ] ] } ]
           [ CT { , WTRSTOP=job_name } ] ]
```

ON	Turns on tracing for a component if the component trace is currently off. If the component trace is on and can be changed, this changes the trace options.
OFF	Turns off tracing for the component. If the component is connected to an external writer, the trace is implicitly disconnected from the writer.
COMP= <i>name</i>	Identifies the component trace with the subsystem ID for the trace address space. This is required for each TRACE command.
PARM= <i>mem</i>	Identifies a member of SYS1.PARM or a data set in the system parmlib concatenation containing the parameters used for tracing. Using a parmlib member enables the operator to initiate the trace, change it, or stop it without a message prompting for parameters.

Parameters specified on the TRACE command override the options specified in the parmlib member. The parameters are described in the section [TRACE Command Reply](#).

WTRSTART= *mem_name*

Identifies the member containing the JCL to invoke an external writer and opens the data sets used by the external writer. The member must be a SYS1.PROCLIB cataloged procedure or a job.

After starting the external writer, use the WTR parameter to connect the component trace to the external writer.

WRAP | NOWRAP NOWRAP instructs the system to stop writing data to a data set when the data set is full. With the WRAP parameter, when the data set or group of data sets is full, new data overwrites the oldest data at the start of the data set or the start of the first data set.

If the WTRSTART parameter on the TRACE CT command specifies NOWRAP, the system uses the primary and secondary extents of the data set or sets. If the WTRSTART parameter specifies WRAP or omits the parameter, the system uses only the primary extent or extents.

WTRSTOP=*job_name* Disconnects the external writer from the component trace and closes the data sets used by the external writer.

jobname is the member name if the source JCL is a procedure, or a job name if defined on a JOB statement within the source JCL.

Before stopping the external writer, turn the component trace off with **TRACE CT,OFF** or disconnect the external writer with **WTR=DISCONNECT**.

TRACE Command Reply

In response to a TRACE CT,ON command without the PARM parameter, the system prompts you to specify the component trace options. Use the REPLY command to respond.

```
R id[ ,ASID=( nnnn [ ,nnnn ]... ) ]
    [ ,CONT | ,END ]
    [ ,JOBNAME=( name [ ,name ]... ) ]
    [ ,OPTIONS=( option [ ,option ]... ) ]
    [ ,WTR={ mem_name | DISCONNECT } ]
```

id Use the same identification number (0-9999) from the message to identify the reply.

ASID=(nnnn [, nnnn] ...)

Specifies the address space identifiers, ASIDs of address spaces used as a filter for tracing. Events in the ASIDs are recorded by the component trace.

The parameter contains a list of 0 to 16 hexadecimal ASIDs separated by commas. An empty ASID list, ASID=(), turns off filtering by address spaces.

In the ASID parameter, list all address spaces to be traced. Address spaces for previous traces are not traced unless listed.

CONT or END The CONT parameter continues the reply on another line. The system issues another reply message. You can then continue the reply and repeat any parameters on the continuation line, except END. Repeated parameters are strung together, they do not overlay each other. The END parameter identifies the end of the REPLY.

CONT or END must be the last parameter on the input line.

JOBNAME=(*name* [, *name*] ...)

Names of jobs used as filters for tracing. Events in these jobs are recorded by the component trace.

The parameter contains a list of 0 to 16 job names separated by commas.

An empty job list, JOBNAME=(), turns off filtering by jobs.

In the JOBNAME parameter, list all jobs to be traced. Jobs specified for previous traces are not traced unless listed.

OPTIONS=(*option* [, *option*] ...)

Specifies the component trace options described in [TRACE Command Reply Options](#).

WTR= *mem_name* | DISCONNECT

The *membername* identifies the member containing the source JCL that invokes the external writer. The member must be a SYS1.PROCLIB cataloged procedure or a job. The membername in the WTR parameter must match the membername in the TRACE CT,WTRSTART command.

WTR=DISCONNECT disconnects the external writer. The component continues tracing and placing the trace records in the address-space buffer, but stops passing trace records to the external writer.

You must also specify a TRACE CT,WTRSTART or TRACE CT,WTRSTOP command to start or stop the writer.

TRACE Command Reply Options

Use the Options parameter in response to the Reply prompt.

```
OPTIONS= [ BUFFERS( size,num ) ] [ BUFFTME( time ) ] [ DATASIZE( dsize ) ]
          [ GROUPS( ( group[, 'filter' ] )... ) ] [ HALT ]
          [ INSTANCE( inst ) ] [ STATUS ] [ TRACE SIZE( tsize ) ] [ WRAP ]
          [ NOWRAP ]
```

BUFFERS (*size,num*) Size of the trace buffers in kilobytes or the number of buffers.

size A value between 64 and 1024.

Default: 256.

num A value between 2 and 128.

Default: Four.

BUFFERS can only be specified for a new trace instance. BUFFERS is optional and should only be specified when creating a trace instance.

Note: If specified when modifying a trace instance, it is ignored.

BUFFTME(*time*) Buffer timeout interval in seconds. At the end of each interval, if the current buffer contains data but is not full, a buffer flush operation is initiated.

BUFFTME is optional and is considered only when creating a trace instance.

Use this parameter to force a buffer switch so you do not have to wait for the entire buffer to fill to see trace data.

Range: 0 – 99999.

Default: 10.

DATASIZE(*dsize*) Specifies the maximum size of a trace record in kilobytes. Trace records that exceed the specified value are truncated.

DATASIZE is optional and can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored. If the specified maximum size exceeds the largest supported trace record size (64 KB less control headers), then the specification has no effect.

Default: Any size record is recorded (up to the maximum IBM limit of 64 KB less control headers.)

GROUPS((*group* [, '*filter*']) ...)

Trace group or groups for which data is collected and optionally a filter parameter for each group.

Use the GROUPS parameter to limit the amount of data collected. Used with INSTANCE to modify an existing trace instance, either adds a new group to the trace instance or replaces an existing group for the trace instance. Once added, a group cannot be removed from the active trace instance.

You can specify a maximum of four trace groups. For details, see the GROUPS parameter description for TCPEEP.

HALT

Specifies that the trace instance identified by the INSTANCE keyword should be stopped.

If HALT is specified, INSTANCE must be specified. If the INSTANCE is connected to an external writer, WTR=DISCONNECT must also be specified.

INSTANCE (*inst*)

Modifies or stops a trace instance.

inst is the component trace address space ID. A new trace instance is created if *inst* is not specified.

STATUS

Displays all active trace instances.

Note: If STATUS is specified, any other keywords are ignored.

TRACESIZE(*tsize*)

Maximum number of trace records to record. If not specified, there is no limit to the number of records recorded.

TRACESIZE is optional and can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

WRAP

Create an in-memory WRAP trace.

Not valid with WTR=*mem_name*, nor can the instance be modified later to specify WTR=*mem_name*. This is the default if NOWRAP is not specified and WTR=*mem_name* is omitted.

NOWRAP

Creates a NOWRAP trace. This is the default if WTR=*mem_name* is specified. The external writer writes the buffers when they fill up and after being written are reused.

This option is for the Trace Address space and should not be confused with WRAP | NOWRAP on the TRACE CT command, which is for the IBM writer.

Comparing TCPEEP and MVS TRACE Syntax

The syntax for commands using TCPEEP differ from those used with the MVS TRACE facility.

OPTIONS

To use MVS TRACE to specify NETIF and PROTOCOL(UDP), you need to specify:

```
OPTIONS = ( GROUPS ( ( TELNET ) ) )
```

This is the same as the TSO TCPEEP command:

```
TCPEEP GROUPS ( ( TELNET ) )
```

JOBNAME

To specify JOBNAME using the MVS TRACE command, the syntax is:

```
JOBNAME = ( jobname1, ..., jobname16 )
```

When using TCPEEP the syntax is

```
TCPEEP JOBNAME ( jobname1, ..., jobname16 )
```

For more information about TCPEEP, see [TCPEEP](#).

Trace Command Examples

The following examples demonstrate the control of the T03 Trace Facility using the MVS TRACE command. The examples assume a T03 Trace Facility is active and is using a Subsystem ID of ACTR.

Starting a Trace Instance

In this example a new trace instance is created. The instance is limited to tracing 10000 records for group ID TELNET from address space RUNTLN.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( GROUPS(TELNET) TRACESIZE( 10000 ) ),JOBNAME=(RUNTLN),END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR909I Trace start successful Instance(01)
```

Modifying a Trace Instance

In this example an existing trace instance is modified. The trace instance is changed to include records for group ID TELNET.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( INSTANCE( 1 ) GROUPS( TELNET ) ),END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR911I Trace modify successful Instance(01)
```

Displaying Trace Status

In this example the status of the trace instances is displayed.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( STATUS ),END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR916I Instance(01) Active, records=23,745
T03TR916I Instance(02) Active, records=1,576
```

Stopping a Trace Instance

In this example an existing trace instance is stopped.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( INSTANCE( 1 ) HALT ),END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR910I Trace shutdown successful Instance(01)
```

Stopping All Trace Instances

In this example all existing trace instances are stopped.

```
TRACE CT,OFF,COMP=ACTR
T03TR910I Trace shutdown successful Instance(01)
T03TR910I Trace shutdown successful Instance(02)
```

Starting an External Writer

In this example an external writer is started. T03XWTR is the name of a predefined started task.

```
TRACE CT,WTRSTART=T03XWTR
```

Starting a Trace Instance and Connecting an External Writer

In this example a new trace instance is created and connected to an external writer. Only one instance can have an external writer connected.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( GROUPS( TELNET ) ),WTR=T03XWTR,END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR909I Trace start successful Instance(01)
```

Modifying a Trace Instance to Connect an External Writer

In this example an existing trace instance is connected to an external writer. Only one instance can have an external writer connected.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( INSTANCE( 1 ) ),WTR=T03XWTR,END
```

Where *xx* ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND. The resulting message is:

```
T03TR911I Trace modify successful Instance(01)
```

Modifying a Trace Instance to Disconnect an External Writer

In this example an existing trace instance is disconnected from an external writer.

```
TRACE CT,ON,COMP=ACTR  
R xx,OPTIONS=( INSTANCE( 1 ) ),WTR=DISCONNECT,END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR911I Trace modify successful Instance(01)
```

Stopping a Trace Instance and Disconnecting an External Writer

In this example an existing trace instance is to be stopped and disconnected from an external writer.

```
TRACE CT,ON,COMP=ACTR  
R xx,OPTIONS=( INSTANCE( 1 ) HALT ),WTR=DISCONNECT,END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR910I Trace shutdown successful Instance(01)
```

Stopping an External Writer

In this example an external writer is stopped. The writer must be disconnected from a trace instance before it can be stopped.

```
TRACE CT,WTRSTOP=T03XWTR
```

Processing Trace Data

In order to process the data collected with the TRACE command, use the IPCS CTRACE command. It handles trace data that is in the MVS Component Trace Entry (CTE) format. By collecting trace data in this format, it is possible to use IPCS facilities to format the trace data. IPCS exit routines need to be written to locate trace records in dumps, filter trace records, and drive trace record formats.

Index

A

abbreviation, verb, 2-12

ABEND, collecting information for Customer Support, 4-4

ALL

- GTF statement parameter, 2-17
- parameter of the MODULE statement, 2-21
- SNAP statement parameter, 2-14

APP

- task group identifier, 2-11

ARM command

- DEREGISTER parameter, 2-16
- ENABLE parameter, 2-15
- LISTEN parameter, 2-15
- NOLISTEN parameter, 2-16
- READY parameter, 2-15
- REGISTER parameter, 2-15
- START parameter, 2-15
- STOP parameter, 2-15
- WAITPRED parameter, 2-15

ARM, dynamic configuration command, 2-15

ATTR, POOL statement parameter, 2-24

B

batch job, TCPEEP, 5-2

C

CB, GTF statement parameter, 2-17

CLASS

- LOGGING statement parameter, 2-19
- SNAP statement parameter, 2-14

CLEAR, P statement parameter, 2-23

CNFG

- START statement parameter, 2-26

command scripts, 2-30

- FLUSH command statement, 2-31
- LIST command statement, 2-31
- NOFLUSH command statement, 2-31
- NOLIST command statement, 2-31

commands

- ARM dynamic configuration, 2-15
- comments, adding, 2-13
- delimiters, 2-14
- format, 2-10
- GTF, IJT system command, 2-17
- keyword operands, 2-13
- keywords, abbreviating, 2-13
- objects, 2-12
- operands, 2-12
- operator, STOP, 2-3, 2-4
- P CLEAR system command, 2-4
- pool information, 2-23
- REFRESH, system command, 2-16
- starting task groups, 2-26
- STATUS, general task group command, 2-15
- system, 2-9
- task group, 2-14
 - execution options, 2-25
 - information, 2-29
 - termination, 2-27
- trace table status, 2-29
- TRACE, TSO diagnostic command, 5-10
- verbs, 2-11
 - DISPLAY, 2-11
 - MODIFY, 2-11
 - VARY, 2-11

COMMANDS, HELP statement parameter, 2-18

comments, adding, 2-13

Component Trace, 5-12

configuration commands

ARM dynamic configuration, 2-15
configuration file, STARTxx member, 2-9
customization, startup JCL, 2-7

D

DECLEN, MEM statement parameter, 2-22
delimiters, 2-14
DEMO, SET statement parameter, 2-25
DEREGISTER
 parameter of the ARM command, 2-16
DEST, LOGGING statement parameter, 2-19
diagnostic commands
 TSO
 GTF, 2-17
 HELP, 2-18
 IFS, 2-18
 LOGGING, 2-18
 MEM, 2-21, 2-22
 MODULE, 2-21
 P, 2-23
 POOL, 2-23
 SET, 2-25
 SRC, 2-26
 START, 2-26
 STC, 2-27
 STOP, 2-27
 SVCDUMP, 2-28
 TASK, 2-29
 TIME, 2-29
 TRACE, 2-29
 VSM, 2-30
diagnostic procedures, 4-1
DISABLE
 parameter of the ARM command, 2-15
DISPLAY
 command verb, 2-11
 GTF statement parameter, 2-17

E

EI, GTF statemen parameter, 2-17
ENABLE
 parameter of the ARM command, 2-15

execution options, 2-25

F

failures
 due to network problems, 4-2
 loop detection, 4-2
 storage shortages, 4-3
 that produce SVC dumps, 4-2
FREE=CLOSE, with logspin, 2-8

G

GENERAL, HELP statement parameter, 2-18
GREETING, REFRESH statement parameter, 2-16
GTF
 command script, 2-31
 IJT system command, 2-17
 SET statement parameter, 2-25
 tracing, 2-17

H

HELP
 IJT system command, 2-18
HEXLEN, MEM statement, 2-22

I

IFS
 IJT system command, 2-18
 MVS statement parameter, 2-22
IJT system commands
 GTF, 2-17
 HELP, 2-18
 IFS, 2-18
 LOGGING, 2-18
 MEM, 2-21, 2-22
 MODULE, 2-21
 P, 2-23
 POOL, 2-23
 SET, 2-25
 SRC, 2-26
 START, 2-26

- STCK, 2-27
- STOP, 2-27
- SVCDUMP, 2-28
- TASK, 2-29
- TIME, 2-29
- TRACE, 2-29
- VSM, 2-30

IJT task group identifier, 2-11

internal tracing, 2-29

Interval record tables

- subtypes 80 and 100, 3-6

J

JCL

- customization startup JCL, 2-7
- descriptions, 2-5
- distribution tape description, 2-5
- startup sample, 2-6

JESCT, MVS statement parameter, 2-22

K

keyword operands

- abbreviating, 2-13
- overview, 2-13

L

LISTEN parameter of the ARM command, 2-15

LNKLST, MVS statement parameter, 2-22

LOGGING statement, implementing logspin utility, 2-8

LOGGING, IJT system command, 2-18

logspin

- implementing with LOGGING statement, 2-8
- T01LOG utility, 2-8

LUPARM, REFRESH statement parameter, 2-16

M

maintenance status, 2-15

MEM, IJT system command, 2-21, 2-22

MEMBER

- START statement parameter, 2-27

MOD

- GTF statement parameter, 2-17
- MEM statement parameter, 2-21

MODIFY

- command verb, 2-11
- GTF statement parameter, 2-17

MODULE, IJT system command, 2-21

MVS

- environment information, 2-22
- SMF record header table, 3-2
- SNAP statement parameter, 2-14

N

NOLISTEN

- parameter of the ARM command, 2-16

NOW, LOGGING statement parameter, 2-19

O

OFF

- GTF statement parameter, 2-17
- SET statement parameter, 2-25
- TRACE statement parameter, 2-29

ON

- GTF statement parameter, 2-17
- SET statement parameter, 2-25
- TRACE statement parameter, 2-29

orderly shutdown, 2-27

P

P CLEAR system command, 2-4

P command, IJT system command, 2-23

performance

- improving with logspin feature, 2-8

POOL

- command options default values, 2-24
- HELP statement parameter, 2-18

- IJT system command, 2-23
- pool, definition of, 2-23
- positional operands, 2-12
- PRINT, LOGGING statement parameter, 2-19
- problems
 - documentation requirements, 4-3
 - incorrect Telnet output, 4-4
 - startup and parameter errors, 4-4

R

- R,TRACE command reply, 5-13
- READY
 - parameter of the ARM command, 2-15
- recycling the TCP/IP stack, 2-4
- REFRESH,system command, 2-16
- REGISTER
 - parameter of the ARM command, 2-15
- report writer
 - commands, 3-9
 - program, JCL, 3-9
- restarting task groups, 2-5
- ROUTCDE, LOGGING statement parameter, 2-19

S

- S0C1 termination at shutdown, 2-4
- S0C4 termination at shutdown, 2-4
- scripts, command, 2-30
- SCVT, MVS statement parameter, 2-22
- SET, IJT system command, 2-25
- shutdown
 - ABENDs, 2-4
 - orderly, 2-27
 - with P CLEAR, 2-4
- SIZE, START statement parameter, 2-29
- SMCA, MVS statement parameter, 2-22
- SMF
 - descriptor section table, 3-2
 - MVS record header table, 3-2

- product section table, 3-3
- report writer commands, 3-9
- report writer program, 3-9
- specification of, 3-1
- standard record header samples, 3-2
- task identification section table, 3-4
- Telnet subtype 23 data section table, 3-4
- user identification section table, 3-4
- virtual storage statistics subtype 80 data section table, 3-6

- SNAP, general task group command, 2-14
- SPIN, LOGGING statement parameter, 2-20
- SPOOL usage with T01LOG logspin utility, 2-8
- SRC
 - IJT system command, 2-26
 - overview, 2-10
- SSCT, MVS statement parameter, 2-22
- START
 - IJT system command, 2-26
 - parameter of the ARM command, 2-15
- starting
 - STARTxx command script, 2-30
 - Telnet, 2-2
- startup, 2-2
- startup errors, 4-4
- STARTxx member
 - command script, 2-30
 - customization, 2-9
 - PARM library, 2-9

- STATUS, general task group command, 2-15
- STCK, IJT system command, 2-27
- STOP
 - IFS operator command, 2-4
 - IJT system command, 2-27
 - operator command, 2-3
 - parameter of the ARM command, 2-15
- subsystem recognition character. SRC
- SVCDUMP, IJT system command, 2-28
- SYSOUT
 - files, T01LOG, 2-8
 - free SYSOUT data set, 2-9
- system commands, 2-9
 - GTF, IJT task group, 2-17
 - HELP, IJT task group, 2-18

- IFS, IJT task group, 2-18
- LOGGING, IJT task group, 2-18
- MEM, IJT task group, 2-21, 2-22
- MODULE, IJT task group, 2-21
- P, IJT task group, 2-23
- POOL, IJT task group, 2-23
- REFRESH, APP task group, 2-16
- SET, IJT task group, 2-25
- SNAP, general task group command, 2-14
- SRC, IJT task group, 2-26
- START, IJT task group, 2-26
- STATUS, general task group command, 2-15
- STCK, IJT task group, 2-27
- STOP of IJT task group, 2-27
- SVCDUMP of IJT task group, 2-28
- TASK of IJT task group, 2-29
- TIME of IJT task group, 2-29
- TRACE of IJT task group, 2-29
- VSM, IJT task group, 2-30

T

T01LOG, logspin utility, 2-8

tables

- JCL descriptions, 2-5
- POOL command options default values, 2-24
- SMF descriptor section, 3-2
- SMF MVS record header, 3-2
- SMF product section, 3-3
- SMF task identification section, 3-4
- SMF Telnet subtype 23 data section, 3-4
- SMF user identification section, 3-4
- SMF virtual storage statistics subtype 80 data section, 3-6

TASK

- IJT system command, 2-29
- REFRESH statement parameter, 2-16

task groups

- commands, 2-14
- execution options, 2-25
- graceful termination, 2-2
- identifier
 - APP, 2-11
 - IJT, 2-11
 - TGI, 2-11
- information, 2-29
- maintenance status, 2-15
- restarting, 2-5
- starting, 2-26
- termination, 2-23, 2-27

TCP/IP stack

- introduction, 1-1
- orderly shutdown, 2-2
- starting, 2-8
- stopping, 2-8

TCPEEP

- batch job, 5-2
- FORMAT, 5-5
- TELNET, 5-6
- TN3270E, 5-6
- TNASCII, 5-5
- TNDATA, 5-5
- TNEBCDIC, 5-5

TELNET

- tracing, 5-6

TEST, SET statement parameter, 2-25

TG, SET statement parameter, 2-25

TGI. *See* task group identifier

TIME

- IJT system command, 2-29

TN3270E, tracing, 5-6

TNASCII, 5-5

TNDATA, 5-5

TNEBCDIC, 5-5

trace

- GTF, 2-17
- internal, 2-29
- table status, 2-29

TRACE

- command, 5-12
- CT, 5-12
- IJT system command, 2-29
- REPLY command, 5-13
- TSO diagnostic command, 5-10

U

USSTAB, REFRESH statement parameter, 2-16

V

VARY, command verb, 2-11

verb abbreviation, 2-12

VSM, IJT system command, 2-30

VTAM major node activation, 2-5

W

WAITPRED

parameter of the ARM command, 2-15

WTO, LOGGING statement parameter, 2-20